Computational Synthesis and Creative Systems

Tony Veale
F. Amílcar Cardoso *Editors*

Computational Creativity

The Philosophy and Engineering of Autonomously Creative Systems

Springer

Computational Synthesis and Creative Systems

Creativity has become the motto of the modern world: everyone, every institution, and every company is exhorted to create, to innovate, to think out of the box. This calls for the design of a new class of technology, aimed at assisting humans in tasks that are deemed creative.

Developing a machine capable of synthesizing completely novel instances from a certain domain of interest is a formidable challenge for computer science, with potentially ground-breaking applications in fields such as biotechnology, design, and art. Creativity and originality are major requirements, as is the ability to interact with humans in a virtuous loop of recommendation and feedback. The problem calls for an interdisciplinary perspective, combining fields such as machine learning, artificial intelligence, engineering, design, and experimental psychology. Related questions and challenges include the design of systems that effectively explore large instance spaces; evaluating automatic generation systems, notably in creative domains; designing systems that foster creativity in humans; formalizing (aspects of) the notions of creativity and originality; designing productive collaboration scenarios between humans and machines for creative tasks; and understanding the dynamics of creative collective systems.

This book series intends to publish monographs, textbooks and edited books with a strong technical content, and focuses on approaches to computational synthesis that contribute not only to specific problem areas, but more generally introduce new problems, new data, or new well-defined challenges to computer science.

More information about this series at http://www.springer.com/series/15219

Tony Veale • F. Amílcar Cardoso
Editors

Computational Creativity

The Philosophy and Engineering of Autonomously Creative Systems

Springer

Editors
Tony Veale
School of Computer Science
University College Dublin
Dublin, Ireland

F. Amílcar Cardoso
Departamento de Engenharia Informática
University of Coimbra
Coimbra, Portugal

ISSN 2509-6575 ISSN 2509-6583 (electronic)
Computational Synthesis and Creative Systems
ISBN 978-3-319-43608-1 ISBN 978-3-319-43610-4 (eBook)
https://doi.org/10.1007/978-3-319-43610-4

This Springer imprint is published by the registered company Springer Nature Switzerland AG
The registered company address is: Gewerbestrasse 11, 6330 Cham, Switzerland

Contents

Chapter 1
Systematizing Creativity: A Computational View

Tony Veale, F. Amílcar Cardoso and Rafael Pérez y Pérez

Abstract Creativity is a long-cherished and widely studied aspect of human behavior that allows us to reinvent the familiar and to imagine the new. *Computational Creativity* (CC) is a recent but burgeoning area of creativity research that brings together academics and practitioners from diverse disciplines, genres and modalities, to explore the potential of our machines to be creative in their own right. As a scientific endeavor, CC proposes that computational modeling can yield important insights into the fundamental capabilities of both humans and machines. As an engineering endeavor, CC claims that it is possible to construct autonomous systems that produce novel and useful outputs that are deserving of the label "creative." The CC field seeks to establish a symbiotic relationship between these scientific and engineering endeavors, wherein the artifacts that are produced also serve as empirical tests of the adequacy of scientific theories of creativity. We argue that, if sufficiently nurtured by volumes such as this, the products of CC research can have a significant impact on many aspects of modern life, with real consequences for the worlds of entertainment, culture, science, education, design, and art.

1.1 From C to CC

Creativity is a multifaceted phenomenon that manifests itself in different guises in different domains. So creativity in the domain of sports (e.g. as manifest in a team sport like soccer, or an intellectual game like chess or Go) is clearly different from

Tony Veale
School of Computer Science, University College Dublin, Ireland.
e-mail: tony.veale@gmail.com

F. Amílcar Cardoso
DEI / CISUC, University of Coimbra, Portugal. e-mail: amilcar@dei.uc.pt

Rafael Pérez y Pérez
Universidad Autónoma Metropolitana, Cuajimalpa, Mexico. e-mail: rpyp@unam.mx

T. Veale, F. A. Cardoso (eds.), *Computational Creativity*, Computational Synthesis and Creative Systems, https://doi.org/10.1007/978-3-319-43610-4_1

creativity in the arts domain (e.g., consider painting or poetry), yet there are enough similarities for exemplary outcomes in each domain to be deserving of the same label, "creative." This heterogeneity makes creativity a notoriously difficult concept to pin down in formal terms, and definitions that favor one area of human activity (such as art) are unlikely to do justice to other areas (such as science, engineering, or cooking). Our definitions of creativity – and a great many have been considered in the scientific literature – are no more than accepted conventions, and it is in the very nature of creativity to bend and subvert these conventions.

Computational Creativity (CC) is an emerging branch of artificial intelligence (AI) that studies and exploits the potential of computers to be more than feature-rich tools, and to act as autonomous creators and co-creators in their own right. In a CC system, the creative impetus should come from the machine, not the human, though in a hybrid CC system a joint impetus may come from both together. As a discipline, CC draws on research in artificial intelligence, cognitive science, psychology, and social anthropology to explore the following questions:

- What does it mean to be "creative"? Does creativity reside in the producer, in the process, in the product, or in a combination of all three together?
- How does creativity relate to expertise and to what extent does it necessitate specialized domain knowledge?
- How does creativity exploit and subvert norms and expectations?
- How are the outputs of creativity judged and evaluated? How can we meaningfully measure creativity? What knowledge is needed (of the creator or process) before we can label a work "creative"?
- What constitutes creativity in different domains and modalities?
- How does creativity emerge from group behavior and collective action?
- What cognitive paradigms offer the most insightful explanatory theories of creativity (e.g., search in a conceptual space, conceptual blending, or bisociation)?

Each of these questions is just as valid in the study of human creativity as it is to the study of machine creativity. What makes CC different is that it adopts an explicitly algorithmic perspective on creativity, and seeks to tie down the study of creative behavior to specific processes, algorithms and knowledge structures. The goal of CC is not just to theorize about the generative capabilities of humans and their machines, but to build working systems that embody these theoretical insights in engineering reality. So CC is both an engineering discipline and an experimental science, in which progress is made by constantly turning insights into applications that can be experimentally tested and evaluated. The purpose of these applications is to create novel artifacts – stories, poems, metaphors, riddles, jokes, paintings, musical compositions, games, etc. – in which a large measure of the perceived creativity is credited directly to the machine. We believe that the future of intelligent computers lies in transforming our computers from passive tools into active co-creators, and that CC is the field that can make this transformation a reality.

CC researchers tend not to trade in definitions of creativity per se, but to focus on those aspects of behavior – in both humans and computers – that produce outputs that are novel or surprising and which yield unexpected value. It was in this vein that

Newell, Shaw, and Simon (1963) suggested four different criteria for categorizing an answer to a question, or a solution to a problem, as "creative":

1. The answer has novelty and usefulness (for the individual or for society).
2. The answer demands that we reject ideas we had previously accepted.
3. The answer results from intense motivation and persistence.
4. The answer comes from clarifying a problem that was originally vague.

Though obviously incomplete, each criterion is instinctively appealing because each expresses in literal language the meaning of a conventional metaphor of creativity. For instance, criterion (1) simply reflects the folk view that creative solutions should be "fresh" and "innovative," perhaps even "ground breaking"; criterion (2) suggests that one must "think outside the box" and reject conventional categories and labels; criterion (3) suggests that to be creative, one must expend copious amounts of "mental energy" in tenaciously exploring the avenues of a wide-ranging conceptual space; and criterion (4) espouses the common belief that creativity requires "illumination" and "insight".

Given the obvious difficulties of distilling a pure definition of creativity – *pure*, at least, in the sense of being metaphor-free and grounded in objective fact rather than in human intuition – CC researchers pursue one or all of the following approaches:

1. They ignore the need to define the phenomenon objectively, and perhaps employ instead an ad hoc definition for convenience; this allows practical work on creative systems to continue, perhaps even to such an extent that practical results can eventually inform a fuller and more satisfying definition of creativity.
2. They embrace the metaphorical foundations of creativity, to identify processes and mechanisms within our repertoire of computational algorithms and representations that best seem to embody these folk metaphors.
3. They identify an archetypal area of creative endeavor and attempt to model that area computationally. In such work, a formal definition is not needed to label the research as "creative." However, as in criterion (1) above, the outputs of this research may then feed back into a later formalization of creativity.

These three alternatives summarize, more or less, the research assumptions made by contemporary CC researchers. Because the field is anchored in engineering and experimentation, CC systems produce concrete outputs whose novelty and value can be assessed by human judges in the absence of any formal definition of creativity. Though many CC researchers believe that machines can exhibit creativity on their own terms, perhaps even by using algorithms and knowledge structures that are different from those used by humans, a principal goal of CC is for machines to exhibit human-level creativity that humans will also perceive as "creative." In striving for this technical goal, CC researchers and their systems can illuminate the processes and biases of human creativity too.

While humans and computers can be creative in the absence of a formal definition of *how* they are being creative, both still need a level of self-understanding and critical awareness to justify the use of the label "creative." Computers which generate outputs for an external user to evaluate are merely generative in their behavior, and

mere generation does not rise to the level of human creativity. Rather, the generation of outputs must be coupled with an awareness of the value of the output in terms of its novelty and its utility. A creative computer must embody a particular view of creativity that the computer itself understands, so that the computer can justify its outputs much as a human creator would do. Such a computer cannot be a dumb savant that naively flings outputs at an audience. Crucially, it must exhibit an ability to filter its outputs for quality, so that any outputs presented to a user show intentionality and discernment, and just as importantly, it must exhibit an ability to articulate why its outputs may have interesting and unexpected value for its audience. Thus, according to the *investment theory* of creativity (Sternberg & Lubart, 1995; Sternberg & Lubart, 1996), a creative computer must be able to articulate its sense of how a particular product or idea can be "bought low and sold high."

Though developments in the field of AI have become fixtures in the technological landscape (e.g., machine translation, natural language question answering, driverless cars, grandmaster-level chess and Go), humans still instinctively cling to the idea that creativity is a uniquely human (or uniquely biological) preserve. In this view, when computers apparently exhibit some measure of creativity, this mere appearance of creativity is due to some specifiable slice of the programmer's own creativity having been imprinted onto the algorithmic workings of the system. In CC research this idea is known as *pastiche*, since such computers unknowingly resort to the same kind of stylistic mimicry that is knowingly exploited by uncreative human artists. For instance, careful musicological analysis of the structure of Bach cantatas can allow a programmer to write software that generates its own novel cantatas in the style of Bach. Though these outputs may fool the human listener, and even delight the unsophisticated ear, they are the product of a system that mimics rather than creates. Such a system has no awareness of its inherent limitations, nor does it have any conceptual input into the hardwired (albeit pseudorandom) processes that it follows. Such systems are more like skilled forgers than creative artists; while they can expertly mimic and recycle, they cannot innovate, nor can they surprise. Moreover, because they explore a predefined sweet-spot in the space of possible outputs, pastiche systems take no risks, always produce well-formed outputs, and have no need to self-critique or to ever learn from their failures.

Of course, pastiche has its place, both in human and in machine creativity. One can learn from pastiche, and even good creators occasionally lapse into pastiche (recognizing this tendency in himself, Picasso once noted of his own paintings "Sometimes I paint fakes"). Pastiche thus serves as a useful boundary case for computational creativity. Indeed, there are cases where pastiche is precisely what the human co-creator desires (e.g., "let's explore more variations on this theme"). Pastiche-based systems are a useful starting point for the computational exploration of creativity, but the goal of CC as a field is to actively transcend pastiche, to demonstrate that computers are capable of true, human-level creativity.

CC is an interdisciplinary research field that sits at the intersection of the fields of AI, psychology, cognitive science, linguistics, anthropology and other human-centered sciences. Given its focus on system building, the field has most in common with AI, and builds on many of the same foundations, such as intelligent search in a

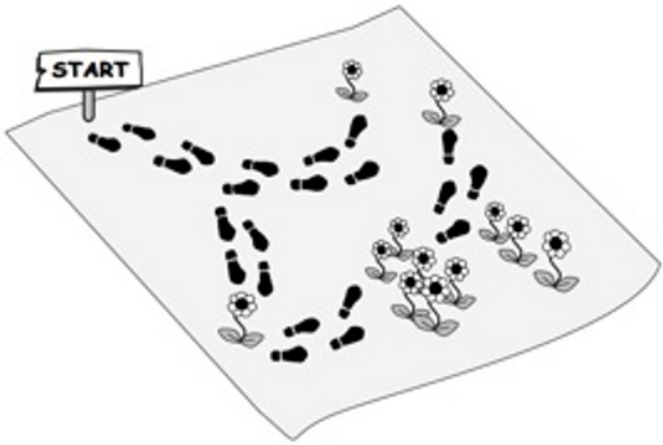


Fig. 1.1 Search in a state space (Veale, 2012). Flowers represent acceptable goal states or solutions, while footprints illustrate the paths pursued via various cognitive agents.

conceptual/problem/state space. Nonetheless, the field has a distinctive character of its own, which shapes its use of ideas and techniques from other fields. For instance, CC views creativity as arising from more than merely a systematic search of a conceptual space of possibilities. Rather, it recognizes that these spaces are deeply-rutted with conventional pathways, and that creativity arises from how an intelligent agent knowingly exploits or subverts these conventions. Thus, Boden (1990) suggests ways in which creativity might arise from the exploration of such a space, while G. Wiggins (2006) has formalized the CC components of this perspective.

A visual representation of a search in a conceptual space is presented in Fig. 1.1. Here, flowers depict acceptable solutions – goal states at which a search can profitably terminate – while footprints illustrate the paths taken by a cognitive agent as it explores the space. Since this model projects a physical search into mental spaces, we can understand "mental agility" as the cognitive equivalent of those qualities that are desirable for an agile physical search. For instance, one often needs to backtrack gracefully when at a dead end, and shift smoothly to an alternate avenue of search. Note that the search metaphor is just that, a metaphor, though it is one that some CC researchers nonetheless resent as overly reductive. However, alternate metaphors for creative choice-making may yet be reducible to the nondeterministic exploration of an abstract space.

Adaptability, in particular, seems to be a salient aspect of creative behavior that can be formalized in terms of search spaces. Boden (1990) offers an intriguing view of adaptive creativity, of a kind that not only delivers surprising solutions to a problem, but that also changes the way we view the problem itself. Boden argues that one should distinguish *exploratory* creativity – of the kind visualized in Figs. 1.1 and 1.2 – from *transformational creativity*. While the former explores the space as it is defined by the problem, looking for previously undiscovered or unappreciated states of unexpectedly high value, the latter actively transforms the space. As illustrated

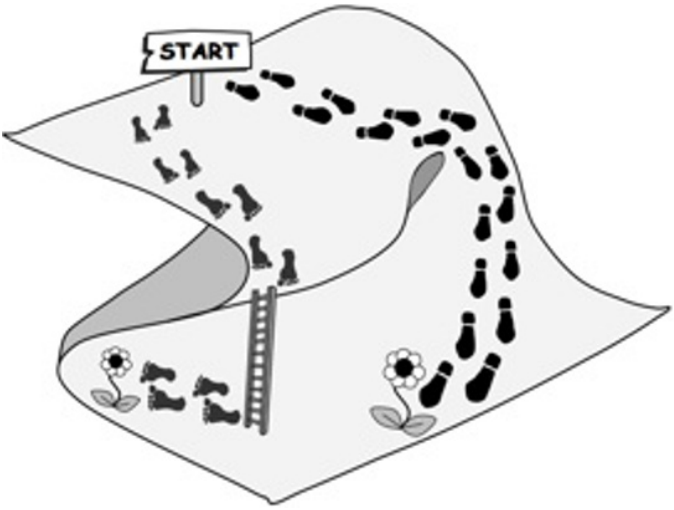


Fig. 1.2 A creative searcher (shown here as a bare-footed explorer) finds novel ways to navigate a search space, for example, by looking in hard-to-reach areas or identifying unconventional connections between states that previously did not appear connected. (Also from Veale (2012))

metaphorically in Fig. 1.3, this transformation redefines the criteria of value that gave shape to the space and which drive the search for value in that space.

Boden cites the development of atonal music as a dramatic example of transformational creativity, and one can also point to key developments in science, such as the transformational shift from a Newtonian (absolute) to an Einsteinian (relativistic) world view, or from a classical (determinate) to a quantum-mechanical (indeterminate) conception of reality. When searching through a space, whether that space is physical or abstract, a searcher can either contort itself to fit the constraints of the space, or contort the space to fit the needs and values of the searcher.

Transformations of the kind analyzed by Boden are the exception rather than the rule in creativity, in either its *small-C* (everyday creativity on a mundane scale) or *big-C* (exemplary creativity on a historical scale) guise. One finds a more commonplace form of agile exploration of a state space in the narrative jokes that are the common currency of social interaction. Jokes exploit the fact that we all navigate through shared state spaces in our everyday lives, to explain the events in the world around us and to understand the behaviors of our friends and colleagues. These shared spaces have well-trodden pathways that correspond to the commonsense norms of conventional thought processes, but these rutted paths do not always offer the quickest or surest routes to a solution. In cases when the best path to a solution is circuitous and nonobvious, mental agility is not a matter of speed but of sure-footedness. The shortest path can sometimes lead to incongruity and failure.

Jokes employ state spaces that have been deliberately warped, so as to fool the unsuspecting explorer into believing that the quickest and most conventional route is also the most intelligent route. In other words, jokes subvert the logic of intelligent

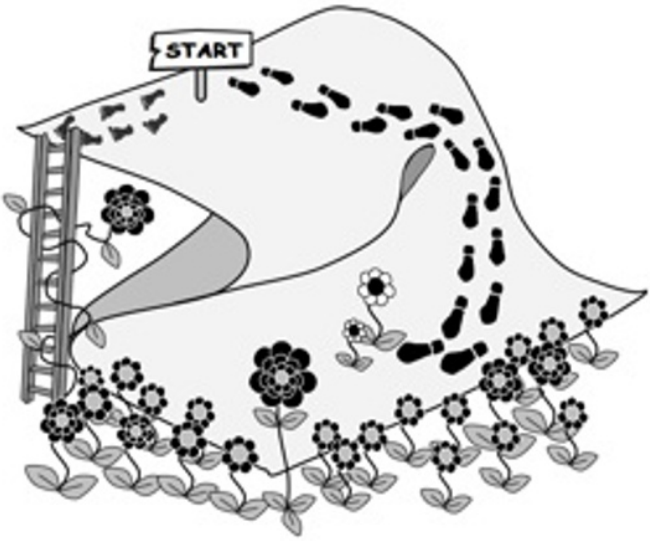


Fig. 1.3 Transformational thinkers alter the space that they are exploring, to identify high-value targets that lie outside the original space, and which would not have been considered in the original formulation of the problem. Of course, a transformation may also put states that were previously accessible out of bounds to the creative agent. (From Veale (2012))

search in a state space, and thereby demonstrate the limits of conventionalized thought processes (Minsky, 1980). The mathematician John Allen Paulos uses the framework of *catastrophe theory* to characterize the kinds of warped spaces that are most used in narrative jokes: as shown in Fig. 1.4, these typically contain an unexpected "kink" or discontinuity that corresponds to a surprising gap in the logic of the narrative (Paulos, 1982; Veale, 2012). Explorers who jump to conclusions by pursuing the path of the discontinuity can be humbled and surprised by their unthinking use of conventional logic.

It is in the computational treatment of discontinuity, incongruity, and contradiction that CC most distinguishes itself from AI as a discipline. In a conventional state space search, contradictions are viewed as dead ends from which a computational agent must backtrack. AI makes an assumption that search is important but the avoidance of search is more important still, so contradictions serve as useful boundaries to limit an otherwise costly search. CC, however, views incongruity and contradiction as opportunities for further search, to explore whether anomalies can be resolved on another level of representation to yield results that are surprisingly meaningful. Resolvable contradictions of this kind underpin not just the incongruity of jokes, but the absurdity of surrealist paintings, the semantic tension of metaphors, the pragmatic insincerity of ironic statements, the plot twists of mystery stories, and even the unexpected discoveries of mathematics and science. In his wide-ranging theory of "Bisociation", Koestler (1964) argued that the creativity in these diverse phenomena emerges from the collision of two seemingly incompatible frames of

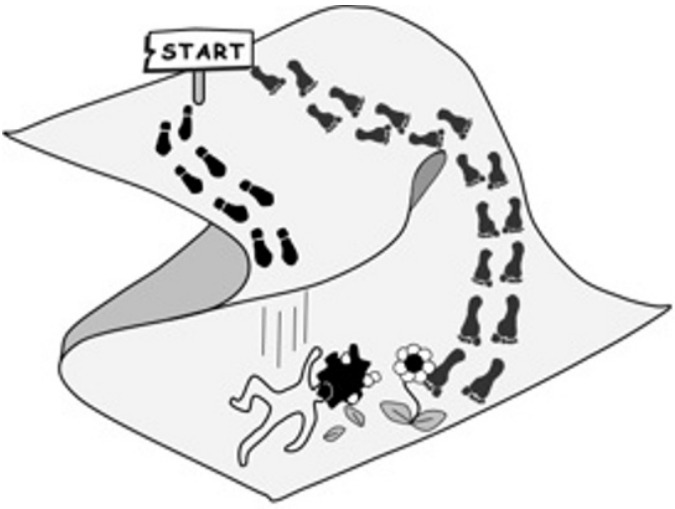


Fig. 1.4 Some state spaces are deliberately constructed to be misleading, and the most obvious or conventional path to the solution can lead to a surprising dead end. A sure-footed explorer who knows the space takes a more circuitous route. (From Veale (2012)).

reference (see also Lavrač et al. (2019) in this volume). Koestler's ideas form the basis of Fauconnier and Turner's influential theory of Conceptual Blending (see Fauconnier (1994) and Fauconnier and Turner (2002)), though it falls mainly to researchers in CC to anchor these ideas in the algorithmic specificity that can only come from computational model building (see e.g., Martins, Pereira, and Cardoso (2019), Pereira (2007), Veale and Li (2011), Veale and O'Donoghue (2000) and Veale (2019)).

1.2 The Association for Computational Creativity

A key development in the history of the field was the establishment of an international Association for Computational Creativity to promote further research in CC and to foster public engagement with the societal issues surrounding CC technologies.

The Association, or ACC, exists to promote the scientific study of human and machine creativity via computational means. The roots of the Association were put down in a series of early workshops and symposia that were explicitly dedicated to the issues of creativity in computers, such as the second Mind conference (1997) and events co-located at AISB (1999-2003), ICCBR (2001), ECAI (2002), EuroGP (2003 and 2004), IJCAI (2003), ECCBR (2004), LREC (2004), IJCAI (2005), and ECAI (2006). Originally, these events were organized by a small cadre of researchers who first coalesced as a working group through EU COST action 282 (Knowledge Exploration in Science and Technology), though this group quickly expanded to

include scientists from diverse parts of the world. In 2007 the community relaunched the International Joint Workshop on Computational Creativity (or IJWCC) as a standalone event and an international steering committee for the workshop and its kindred events was formally established (the history of the IJWCC series is described in a special issue of *AI Magazine* on CC (Cardoso, Veale, & Wiggins, 2009).

In 2008 the steering committee took the decision to transform the workshop into a conference, thus establishing the International Conference on Computational Creativity (ICCC). The first ICCC was held in 2010 in Lisbon, the second in 2011 in Mexico City, the third in 2012 in Dublin, the fourth in 2013 in Sydney, the fifth in 2014 in Ljubljana, the sixth in 2015 in Park City, the seventh in 2016 in Paris, and the eighth in 2017 in Atlanta. The ninth is planned for 2018 in Salemanca, Spain. During the first year of the ICCC in 2010, the members of the steering committee formally recognized the necessity of creating an international Association for Computational Creativity. Thus, in 2011 the Association was officially founded, and Geraint Wiggins was elected as its first Chair. In 2015 Rafael Pérez y Pérez was elected as its next chair, and that same year the ACC's official constitution was ratified by its steering committee.

From its origins to the present day, the Association has pursued a range of community-building activities, from the annual organization of the ICCC to the creation and maintenance of a comprehensive Wikipedia entry on computational creativity and the publication of special journal issues on CC (such as *Knowledge-Based Systems* **9**(7), 2006; *New Generation Computing* **24**(3), 2006; and *Minds and Machines* **20**(4), 2010). The main event organized by the Association is its annual conference, the ICCC. The conference's main goals are to provide a space where researchers from across the world can meet to debate ideas, hear about novel approaches to the study of creativity, build partnerships, and start collaborations that explore interdisciplinary opportunities. The participation of students and young researchers have always been a priority for the ACC. From the beginning, the members of the steering committee have made it their mission to make publicly available all the materials generated by the Association. The proceedings of its past conferences can be downloaded from the Association's web page, www.computationalcreativity.net.

The support of the Association has provided fertile ground for new CC initiatives. In 2013, for example, seven European CC researchers (and members of the Association) obtained support from the European Commission to organize, under the aegis of the PROSECCO coordination action, a range of activities to stimulate enhanced CC outreach and education, including tutorials, summer schools, code camps, workshops, and contact fora. PROSECCO has grown within the environment cultivated by the ACC, just as the Association has itself been shaped and advanced by PROSECCO. For example, the charter of the ACC was a specific deliverable of the PROSECCO project in its first year of operation. Another outcome of this symbiotic relationship is the volume you are now reading, which has always been a planned effort of the ACC but which has now been made a reality as a PROSECCO deliverable. Though PROSECCO's funded lifetime as an EC project ended in late 2016, its legacy will live on in the Association.

The Association faces important challenges in the coming years, the most notable of which is its consolidation as an international society for all CC researchers that will continue to promote the goals, the philosophy, and the technological vision of CC. This consolidation and growth will be only achieved through the committed participation of all of its members.

1.3 The PROSECCO Vision

PROSECCO is an international coordination action that was funded by the European Commission from 2013 to 2016 to "Promote the Scientific Exploration of Computational Creativity." The action was anchored in the belief that our computers can be more than mere "tools" of human creativity, and can actually rise to the level of co-creators that proactively share the creative responsibility with a human peer. As co-creators, our computers will be capable both of generating their own ideas and of framing those ideas in the appropriate modalities (e.g., language, image, sound). The PROSECCO vision of a co-creator is more than a mere facilitator or enabler for human creativity (in the sense, e.g., that Microsoft Word or Adobe Photoshop facilitates content creation, or in the sense that Facebook facilitates collaborative creation), but instead envisions a largely autonomous agent that explores its own conceptual spaces and expresses its own ideas in its own terms. CC conducts application-driven research into this notion of a computational co-creator in two guises: autonomous systems that receive little or no human input, and semiautonomous hybrid systems that interact with humans as peers.

This view is shaped not by a desire to replace humans with machines, nor by a perceived lack of human creativity in modern society, but by the belief that large amounts of human creativity remain untapped because users lack the appropriate co-creation software. No matter how richly featured a conventional software tool may be, users are still forced to start from a blank page or an empty screen, or a predetermined template that simply encourages recycling and pastiche. Future CC systems must not only suggest ideas to users, but also articulate, demonstrate and critique those ideas as would a human teammate. By providing humans with partners that can share the creative responsibilities and the creative credit, the goal is not to replace human creativity, but to engage and foster human creativity as only a creative equal can.

This is an ambitious vision that will take decades to fully realize, though in the interim, researchers in the CC field continue to build systems that serve useful (and steadily improving) creation and co-creation roles. To meet these challenges, CC must go from being a growing area of niche interest to being a true scientific discipline in its own right. The challenges are both organizational and research-based. As a field, it must coalesce around a clear set of principles, an unambiguous and comprehensive terminology, and a canonical set of techniques, metrics and approaches to evaluation (see e.g., Ritchie (2007)). We must consolidate our own identify as a field while actively engaging with neighboring disciplines. Computational Creativity has long

been an implicit element of AI research, one that comes to the fore when AI addresses topics of an obviously creative bent, such as painting (e.g. Harold Cohen's Aaron), analogical reasoning (see e.g., Gentner (1983), Goel (2019), Hofstadter (1995), Veale (2006), Veale and Keane (1997), Winston (1980)), music generation (e.g. the EMI of Cope (2006)), story telling (e.g. the TALE-SPIN system of Meehan (1981) and the MINSTREL system of Turner (1994)), joke generation (e.g. see Binsted, Pain, and Ritchie (1997), Gatti, Ozbal, Guerini, Stock, and Strapparava (2019), Hempelmann (2008)), or metaphor processing (e.g. Fass and Wilks (1983), Veale and Keane (1992), Veale, Shutova, and Klebanov (2016), Wilks (1978)). However, this work was generally seen as AI work, and not as a product of a specific movement toward the realization of true computational creativity.

A key pillar of the PROSECCO coordination action has thus been its educational program, which has sought to inform and shape the next generation of CC researchers. To this end, the project has organized a major tent-pole educational event in each of its three years. Beginning with an Autumn School in 2013, the project organized a code camp in both 2015 and 2016, with a variety of smaller events (such as targeted tutorials) spread between these tent-poles. It was of the utmost importance that student participants at these educational events should not fall into the beguiling trap of *mere generation* – the alluring belief that machines can be programmed to generate creative outputs without being able to appreciate those outputs for themselves – but should instead build generative systems that would be accepted as creative by the CC community. Our machines cannot appreciate their own outputs if they lack knowledge about the semantic components of their outputs: hence there is a need to provide CC students and researchers with a comprehensive knowledge base of interconnected and semantically grounded beliefs. An important outcome of the PROSECCO project has been the development of large-scale semantic resources for use in teaching CC principles and fostering future CC research. Specifically, PROSECCO has developed two complementary semantic knowledge bases to support these goals. The first is the *NOC list*, or Non-Official Characterization list, which provides vivid semantic detail about a large cast of famous personalities (800 at the last count) and their many attributes. The second is the *Scéalextric* knowledge-base of plot structures and idiomatic renderings of story actions, which significantly lowers the otherwise formidable barriers to entry to the CC domain of automated story generation. Each of these resources, which collectively run to over 60,000 high-quality semantic triples, can be accessed on the dedicated PROSECCO GitHub site: github.com/prosecconetwork. In addition, readers may be interested in reading about the experiences of PROSECCO code-camp participants in a blog dedicated to these ongoing educational efforts: bestofbotworlds.com.

This volume of canonical papers constitutes another key part of PROSECCO and the ACC's efforts to reach future and emerging researchers in CC while they are still in the development stages of their education. As an emerging field, CC needs to reach graduate students in a variety of fields and disciplines to create the next wave of active researchers. By reaching these students at a time when their Ph.D. plans are still at a formative stage, this book can bring the necessary knowledge of mathematics, psychology, anthropology, sociology, art, language, music, science, and

design into the field, and demonstrate that CC is a research area in which its students can pursue a truly cross-disciplinary exploration of creativity at the intersection of experimental science, system-building engineering, and the humanities.

1.4 A Thematic Overview

This volume brings together a diversity of papers on a diversity of themes, to collectively chart the terrain that is Computational Creativity. This section provides a brief introduction to each of the chapters and themes that await the readers of the volume.

Novelty is a pillar of many operational definitions of creativity, but what exactly do we mean by "novel"? Is an artifact novel to the extent it differs from others that we have experienced in the past? But difference is itself a contextual notion, since the dimension along which two things can differ will be primed by our expectations of how they should be the same. Kazjon Grace and Mary Lou Maher thus argue here that it more meaningful to say that an artifact is novel to the extent that it violates our expectations of "more of the same" (Grace & Maher, 2019). Expectations shape our perception of novelty and creativity, but these authors also argue that expectations can crucially shape the generation process too.

Novelty is just one dimension along which the "creativity" of an artificial generative system may be evaluated. Anna Jordanous takes a wide-angle look at the issue of evaluation in this volume, to consider the issue from a historical, a strategic and a methodological perspective (Jordanous, 2019). What does it mean to say that a CC system has undergone an evaluation, how might we interrogate the results of an evaluation, and how might we compare two systems that putatively aim to generate the same kinds of artifact? As Jordanous notes, the aim of CC as a field should be to provide a sound and systematic basis for the rigorous evaluation of our automated systems.

This is an aim that also provides the guiding theme for Graeme Ritchie's contribution to this volume. Ritchie argues that growth in the engineering sophistication of CC systems must be matched by comparable growth in the objective rigor and sophistication of our methods for evaluating these systems (Ritchie, 2019). For Ritchie, a proper evaluation shows an understanding of the goals of the work being evaluated, requiring an evaluator to tease apart the theoretical agenda from its engineering application. While noting that CC systems operate in a realm that supports little in the way of objectively defined and widely agreed criteria, Ritchie uses an analysis of past evaluations to establish a solid foundation for the evaluation of future CC systems.

Creative systems can operate in the various modalities that we associate with human creativity, from the visual (e.g., painting, design, video games) to the musical to the linguistic. Each modality is associated with its own sense of what constitutes creative "genius." For language, this sense integrates notions of wit, concision and persuasive power. In their contribution to this volume, Lorenzo Gatti, Gözde Özbal, Marco Guerini, Oliviero Stock, and Carlo Strapparava explore linguistic creativity as

it inheres in puns and in advertising slogans (Gatti et al., 2019). While the former allows CC to build models of the whimsical creativity that we associate with children, the latter – relying on many of the same techniques – allows CC to investigate an area of human activity with a compelling commercial use case.

Storytelling is a capability that defines us as human beings. We use stories in every aspect of our social and intellectual lives, to explain the world both to ourselves and to others. What would otherwise be a mere sequence of events, one occurring after the other, becomes a coherent narrative in the hands of a storyteller. Though some make a livelihood from this ability, we are all natural storytellers, placing narrative at the heart of computational creativity too, not least because so many of our definitions of creativity are little more than compressed narratives of what creativity *should* be. Insofar as we can define creativity at all, and define the social expectations of how a creative person should act, it is because we can tell a good story that draws these expectations into a recognizable narrative. The contribution of Rafael Pérez y Pérez to this narrative in this volume is a model of storytelling (his MEXICA system) that aims to capture how real people tell real stories in social situations (Pérez y Pérez, 2019). As with the best gossip, these stories hinge on how others both obey and subvert social expectations, so Pérez y Pérez sets out in this chapter to capture the kinds of commonsense knowledge that influence our understanding of social norms and those who obey, bend or break them.

As a CC researcher, Pablo Gervás is as much known for poetry generation as he is for story generation. This duality is not a coincidence, but arises from his conviction that memorable poems typically have a memorable story to tell too. Gervás has thus studied the norms of storytelling and the norms of poetry side by side. So, while his contribution to this volume principally concerns the latter, we invite readers to keep the former in mind when enjoying his chapter (Gervás, 2019). There are other points of similarity between this chapter and others in the volume too, since Gervás sets out to evaluate just *how* we should evaluate our computer poets before returning to the subject of his own automated Spanish poet.

Social convention is the invisible hand that guides both the generation and our appreciation of creative artifacts, whether or not we explicitly set out to obey or subvert this conventionality. As CC researchers we often aim to codify the governing conventions for our genre or domain into our systems so that they might satisfy the tastes of a user base entrained in those conventions. Rob Saunders, is his contribution to this volume, uses a multiagent framework to explore how those social conventions might arise in the first place (Saunders, 2019). His artificial agents provide a revealing sandbox in which we can observe the emergence, via self-organization, of norms in creative fields. For Saunders, creativity is not a quality of a lone system, but of a social agent interacting in a dynamic world.

If creativity resides as much in social interactions as it does in algorithmic action, perhaps other complex dynamical systems that exhibit an equivalent sensitivity to interaction and context might also be usefully labeled “creative”? Though we naturally take a human-centric view of creativity even in the context of CC, the core ideas of CC can be observed on much longer and much shorter time scales than the human lifespan, and at biological levels that are much higher and much

lower than the human organism. In this contribution to this volume, Jon McCormack concerns himself with how ideas of a "biological" creativity can guide and inspire – via appropriate acts of abstraction, simplification, and generalization – work in computational creativity (McCormack, 2019).

What is creative in one place or time may not be creative in another. Our concept of creativity, just like our concept of art, changes with the context. One need only look to the history of art, and to the seismic changes that erupt in each new century, to see that our criteria for evaluating a creative artefact are far from static. Just as it is useful to view some artists as the spiritual offspring of those that went before, combining the best traits of earlier pioneers, it is also useful to think of computer-generated art as an evolutionary process. The contribution of João Correia, Penousal Machado, Juan Romero, Pedro Martins, and F. Amílcar Cardoso (2019) to this volume does just this. Their work, which exploits the paradigm of genetic algorithms, assumes that quality is a moving target that a system must be nimble enough, and responsive enough, to follow to the most creative results.

Abstraction and generalization are the guiding themes of Geraint Wiggins's contribution to this volume (G. A. Wiggins, 2019). Wiggins sets out to formalize the intuitions that hold sway in many discussions of creativity that implicitly see it as an exploration in a space of conceptual possibilities. These intuitions are often given a metaphorical form in layman's language, as when we speak of "exploring all avenues" and "coming up empty," of "hitting a brick wall" or "going around an obstacle," of "reaching a dead end, " or of "finding a goldmine of opportunities". Wiggins does more than recast these intuitions in formal language for the sake of formalization – he demonstrates that when expressed in wholly formal terms the ideas allow themselves to be manipulated in ways that are both enlightening and productive.

An unspoken aspect of the exploration view of creativity is that the explorer is an intentional agent, one that explores a space of possibilities with a specific goal (or at least a specific meta-goal) in mind. In his contribution to this volume, Dan Ventura considers the related questions of system *intentionality* and system *autonomy*, both at a philosophical level and at a practical level afforded by his CC system DARCI contributions (Ventura, 2019). Papers such as Ventura's show CC to be a discipline that sits comfortably at the crossroads of philosophy and engineering, where profound questions can be not just asked but answered in practical implementation terms.

Ofttimes CC allows these profound questions to be rephrased in simpler terms so that they might give rise to robust, scalable and useful implementations. In his contribution to this volume, Tony Veale explores the phenomenon of conceptual blending (Veale, 2019), but finds it too powerful and too operationally vague to be properly and faithfully implemented by any real CC system. He identifies a sub-species of conceptual blend that it is more amenable to robust computational modeling, coining the term "conceptual mash-up" to distinguish this related notion from its complex forebear. Importantly, Veale shows that such mash-ups are more than curtailed blends; they capture an important aspect of conceptual blending in a form that allows blends to be more than intellectual curiosities or mere playthings, so that mash-ups can actively fill the gaps in a CC system's representation of the world.

A fuller treatment of the merits of blending is offered by the contribution of Pedro Martins, Francisco C. Pereira and Amílcar Cardoso to this volume (Martins et al., 2019). Taking a historical perspective on the development and subsequent revision of an early CC implementation of conceptual blending – named *Divago* – the authors show how those early systems can provide a sound foundation for building a robust modern approach to the most challenging aspects of human creative behavior. As a field that prizes practical and continuous implementation it can be tempting to view the CC terrain as a junkyard of once-promising but now-abandoned systems and approaches. However, as this chapter (and volume as a whole) shows, CC is a field that gives rise to families of related systems and approaches to knowledge representation that grow and evolve in interesting ways over time.

Indeed, a practical CC system will rely as much (if not more) on its domain knowledge and on a felicitous representation of such as it will on any special algorithms for manipulating this knowledge. For instance, if it is the goal of a CC system to find novel insights at the boundaries of two disparate domains, any effective search will hinge crucially on the construction and representation of those domains. In their contribution to this volume, Nada Lavrač, Matjaz Juršič, Borut Sluban, Matic Perovšek, Senja Pollak, Tanja Urbančič, and Bojan Cestnik focus on scientific knowledge discovery at the overlap of domains that are constituted by different textual subsets of the scientific literature. In this way these authors give a robust statistical form to the intuitions of Arthur Koestler that creativity arises from the *bisociation* (i.e., simultaneous *bi-association* with two domains of knowledge) of two different perspectives or frames of mind (Lavrač et al., 2019).

Last but not least, the domain of choice for Ashok Goel in his contribution to this volume is a domain that encompasses many others, the *design* domain (Goel, 2019). In particular, Goel applies model-based analogical reasoning to the solution of creative problems of design, showing how past engineering solutions can be retrieved and adapted to suit new needs in new contexts. But those past solutions need not be the solutions of human engineers, and Goel shows how model-based analogies can turn *all* of nature into a case-base for biologically inspired design.

1.5 Conclusion: Baby Steps in the Right Direction

The pioneering 19th-century scientist Michael Faraday was once pointedly asked by Benjamin Disraeli about the practical uses of research in the nascent field of electricity. Faraday retorted “What use is a baby?” though in another equally apocryphal telling, Faraday replied “Why, one day you will tax it, sir.” Disraeli’s question seems rather short-sighted with the benefit of hindsight, knowing what we know now about the utility of electricity in modern society. Nonetheless, Faraday had a responsibility as a scientist to educate the general public about his ambitious vision for this startling new phenomenon. The same holds true for the champions of any transformational discipline, and so a dual impact of PROSECCO has been a raised public awareness

of the benefits of creative computers and the shaping of realistic public expectations of progress in the field.

Consider the following response to a debate about the value or otherwise of computer-generated paintings in a BBC documentary about AI and CC. The response was published in a British broadsheet newspaper, *The Daily Telegraph*, after an airing of the documentary :

> . . . one man is trying to teach computers to paint. One picture with colourful dancers was lauded as a creative breakthrough but was actually atrocious. Which proves that as long as computer scientists have no artistic taste, it's unlikely computers ever will. (Daily Telegraph, April 4th, 2013)

The key point is not that the journalist above was *wrong* – the above response is a valid *subjective* response to the output of a CC system – but that the critique was made in an inappropriate frame of reference. CC is a developing discipline and needs to be nurtured as such; expectations must be realistically shaped so that incremental breakthroughs are not cynically strangled at birth, and so that the public can appreciate the merits of computer-generated artifacts as the results of research in progress rather than finished research. The CC community must continue to engage with the public about the merits and possibilities of its research, to refute misconceptions and to respond to genuine concerns. By impacting directly on public expectations, our research can foster an environment in which less-than-human computational creativity can make its way into steadily improving software that is aimed at the general public. This volume, with its diverse collection of contributions, is intended to help future CC researchers to make this transition a practical reality.

Acknowledgements This work was supported by the EC project PROSECCO, a coordination action funded to *PROmote the Scientific Exploration of Computational Creativity.* The official PROSECCO website is: PROSECCO-network.eu. For introductory teaching material on Computational Creativity, visit the PROSECCO-sponsored website *RobotComix.com*

References

Binsted, K., Pain, H., & Ritchie, G. (1997). Children's evaluation of computer-generated punning riddles. *Pragmatics and Cognition*, *5*(2), 309–358.

Boden, M. (1990). *The creative mind: Myths and mechanisms (2nd edition)*. Routledge.

Cardoso, A., Veale, T., & Wiggins, G. (2009). Converging on the divergent: The history (and future) of the international joint workshops in computational creativity. *AI Magazine*, *30*(3), 15–22.

Cope, D. (2006). *Computer models of musical creativity*. Cambridge, MA: MIT Press.

Correia, J., Machado, P., Romero, J., Martins, P., & Cardoso, F. A. (2019). Breaking the mould: An evolutionary quest for innovation through style change. In T.

Veale & F. A. Cardoso (Eds.), *Computational creativity: The philosophy and engineering of autonomously creative systems* (pp. 353–399). Springer.

Fass, D., & Wilks, Y. (1983). Preference semantics, ill-formedness, and metaphor. *Computational Linguistics*, *9*(3-4), 178–187.

Fauconnier, G. (1994). *Mental spaces: Aspects of meaning construction in natural language*. Cambridge University Press.

Fauconnier, G., & Turner, M. (2002). *The Way We Think: Conceptual Blending and the Mind's Hidden Complexities*. New York: Basic Books.

Gatti, L., Ozbal, G., Guerini, M., Stock, O., & Strapparava, C. (2019). Computer-supported human creativity and human-supported computer creativity. In T. Veale & F. A. Cardoso (Eds.), *Computational creativity: The philosophy and engineering of autonomously creative systems* (pp. 235–252). Springer.

Gentner, D. (1983). Structure-mapping: A theoretical framework. *Cognitive Science*, *7*(2), 155–170.

Gervás, P. (2019). Exploring quantitative evaluations of the creativity of automatic poets. In T. Veale & F. A. Cardoso (Eds.), *Computational creativity: The philosophy and engineering of autonomously creative systems* (pp. 273–302). Springer.

Goel, A. (2019). Revisiting design, analogy, and creativity. In T. Veale & F. A. Cardoso (Eds.), *Computational creativity: The philosophy and engineering of autonomously creative systems* (pp. 139–156). Springer.

Grace, K., & Maher, M. L. (2019). Expectation-based models of novelty for evaluating computational creativity. In T. Veale & F. A. Cardoso (Eds.), *Computational creativity: The philosophy and engineering of autonomously creative systems* (pp. 193–207). Springer.

Hempelmann, C. (2008). Computational humor: Beyond the pun? In V. Raskin (Ed.), *The primer of humor research* (pp. 333–360). Berlin: Mouton de Gruyter.

Hofstadter, D. (1995). *Fluid concepts and creative analogies: Computer models of the fundamental mechanisms of thought*. HarperCollins.

Jordanous, A. (2019). Evaluating evaluation: Assessing progress and practices in computational creativity research. In T. Veale & F. A. Cardoso (Eds.), *Computational creativity: The philosophy and engineering of autonomously creative systems* (pp. 209–234). Springer.

Koestler, A. (1964). *The act of creation*. London, UK: Penguin Books.

Lavrač, N., Juršič, M., Sluban, B., Perovšek, M., Urbančič, T., & Cestnik, B. (2019). Bisociative knowledge discovery for cross-domain literature mining. In T. Veale & F. A. Cardoso (Eds.), *Computational creativity: The philosophy and engineering of autonomously creative systems* (pp. 119–138). Springer.

Martins, P., Pereira, F. C., & Cardoso, F. A. (2019). The nuts and bolts of conceptual blending: Multi-domain concept creation with Divago. In T. Veale & F. A. Cardoso (Eds.), *Computational creativity: The philosophy and engineering of autonomously creative systems* (pp. 91–118). Springer.

McCormack, J. (2019). Creative systems: A biological perspective. In T. Veale & F. A. Cardoso (Eds.), *Computational creativity: The philosophy and engineering of autonomously creative systems* (pp. 327–352). Springer.

Meehan, J. (1981). Tale-spin. In R. Schank & C. Riesbeck (Eds.), *Contemporary approaches to creative thinking* (pp. 197–226). Hillsdale, NJ: Lawrence Erlbaum.

Minsky, M. (1980). *Jokes and the logic of the cognitive unconscious (memo no. 603)*. Massachusetts Institute of Technology A.I. Laboratory.

Newell, A., Shaw, J., & Simon, H. (1963). The process of creative thinking. In G. T. H. E. Gruber & M. Wertheimer (Eds.), *Contemporary approaches to creative thinking*. New York: Atherton.

Paulos, J. A. (1982). *Mathematics and humor*. University of Chicago Press.

Pereira, F. C. (2007). *Creativity and artificial intelligence: A conceptual blending approach*. Berlin: Walter de Gruyter.

Pérez y Pérez, R. (2019). Representing social common-sense knowledge in MEXICA. In T. Veale & F. A. Cardoso (Eds.), *Computational creativity: The philosophy and engineering of autonomously creative systems* (pp. 253–272). Springer.

Ritchie, G. (2007). Some empirical criteria for attributing creativity to a computer program. *Minds and Machines*, *17*(1), 67–99.

Ritchie, G. (2019). The evaluation of creative systems. In T. Veale & F. A. Cardoso (Eds.), *Computational creativity: The philosophy and engineering of autonomously creative systems* (pp. 157–192). Springer.

Saunders, R. (2019). Multi-agent based models of social creativity. In T. Veale & F. A. Cardoso (Eds.), *Computational creativity: The philosophy and engineering of autonomously creative systems* (pp. 303–325). Springer.

Sternberg, R., & Lubart, I. T. (1995). *Defying the crowd: Cultivating creativity in a culture of conformity*. Hillsdale, NJ: Lawrence Erlbaum.

Sternberg, R., & Lubart, T. I. (1996). Investing in creativity. *American Psychologist*, *51*(7), 677–688.

Turner, S. R. (1994). *The creative process: A computer model of storytelling*. Hillsdale, NJ: Lawrence Erlbaum.

Veale, T. (2006). Re-representation and creative analogy: A lexico-semantic perspective. *New Generation Computing*, *24*, 223–240.

Veale, T. (2012). *Exploding the creativity myth: The computational foundations of linguistic creativity*. London, UK: Bloomsbury.

Veale, T. (2019). From conceptual mash-ups to bad-ass blends: A robust computational model of conceptual blending. In T. Veale & F. A. Cardoso (Eds.), *Computational creativity: The philosophy and engineering of autonomously creative systems* (pp. 71–89). Springer.

Veale, T., & Keane, M. (1992). Conceptual scaffolding: A spatially founded meaning representation for metaphor comprehension. *Computational Intelligence*, *8*(3), 494–519.

Veale, T., & Keane, M. (1997). The competence of sub-optimal structure mapping on 'hard' analogies. In *Proceedings of the 15th international joint conference on artificial intelligence*, Nagoya, Japan: Morgan Kaufmann.

Veale, T., & Li, G. (2011). Creative introspection and knowledge acquisition: Learning about the world thru introspective questions and exploratory metaphors. In

Proceedings of the 25th conference of the association for the advancement of artificial intelligence, San Francisco: AAAI press.

Veale, T., & O'Donoghue, D. (2000). Computation and blending. *Cognitive Linguistics*, *11*(3-4), 253–281.

Veale, T., Shutova, E., & Klebanov, B. B. (2016). *Metaphor: A computational perspective*. USA: Morgan Claypool: Synthesis Lectures on Human Language Technologies.

Ventura, D. (2019). Autonomous intentionality in computationally creative systems. In T. Veale & F. A. Cardoso (Eds.), *Computational creativity: The philosophy and engineering of autonomously creative systems* (pp. 49–69). Springer.

Wiggins, G. (2006). Searching for computational creativity. *New Generation Computing*, *24*(3), 209–222.

Wiggins, G. A. (2019). A framework for the description, analysis and comparison of creative systems. In T. Veale & F. A. Cardoso (Eds.), *Computational creativity: The philosophy and engineering of autonomously creative systems* (pp. 21–48). Springer.

Wilks, Y. (1978). Making preferences more active. *Artificial Intelligence*, *11*(3), 197–223.

Winston, P. (1980). Learning and reasoning by analogy. *Communications of the ACM*, *23*(12), 689–703.

Chapter 2
A Framework for Description, Analysis and Comparison of Creative Systems

Geraint A. Wiggins

Abstract I summarise and attempt to clarify some concepts presented in and arising from Margaret Boden's descriptive hierarchy of creativity (Boden, M. A., The Creative Mind: Myths and Mechanisms, 2nd Ed., Routledge, London, 2004), by formalising the ideas she proposes. The aim is to move towards a model which allows detailed comparison, and hence better understanding, of systems, whether artificial, natural or hybrid, which exhibit behaviour which would be called 'creative' in humans. The framework paves the way for the description of naturalistic, multi-agent creative artificial intelligence systems, which create in a societal context. I demonstrate some simple reasoning about creative behaviour based on the new framework, to show how it might be useful for the analysis and study of creative systems. In particular, I identify some crucial properties of creative systems, in terms of the framework components, some of which may usefully be proven a priori of a given system. Finally, I exemplify the use of the framework by broadly describing the development of art music from the 10th to the 20th century in these more formal terms.

2.1 Introduction

One of the few attempts to address the problem of creative behaviour in the early days of artificial intelligence (AI) was that of Margaret Boden, perhaps best summarised in her book *The Creative Mind* (Boden, 2004). A common criticism of Boden's approach is that it is rather lacking in detail, and that it is not clear how the various components fit together to give a real account of creative behaviour. It is the aim of

Geraint A. Wiggins
Computational Creativity Lab, School of Electronic Engineering and Computer Science, Queen Mary University of London, Mile End Road, London E1 4FZ, UK.
e-mail: geraint.wiggins@qmul.ac.uk

T. Veale, F. A. Cardoso (eds.), *Computational Creativity*, Computational Synthesis and Creative Systems, https://doi.org/10.1007/978-3-319-43610-4_2

this chapter to give a more uniform and formal treatment of the ideas, which can then be used in a precise way.

Boden's ideas have been debated at some length (Boden, 1995; Haase, 1995; Lustig, 1995; Perkins, 1995; Ram, Wills, Domeshek, Nersessian, & Kolodner, 1995; Schank & Foster, 1995; Turner, 1995), but little attempt has been made to give a mechanism through which they can be applied formally (and thence automatically). In the current chapter, rather than entering into the debate above, I will attempt to make Boden's descriptive hierarchy more precise. In doing so, I will suggest some additions to the theory, which may or may not be implicit in Boden's account, and show how some of the distinctions over which she has been challenged may perhaps be supported. The formalisation will also make it possible to identify desirable and undesirable properties of creative systems in abstract terms.

Given the formulation, I define some crucial properties of (artificial) creative systems in terms of the framework, including some which might be proven a priori and some which may be usefully detectable during the activity of the system.

An earlier version of the framework (Wiggins, 2006a) has been used by Gervás (2002a, 2002b). Gervás demonstrated very clearly that the application of such frameworks is not simply objective but may be varied, depending on the viewpoint and aims of the person doing the applying. This is not a disadvantage: the framework is a philosophical tool to enable discussion and comparison, so making such subjective positions explicit is an important first step, and it is not impossible that there may be more than one valid way to analyse a given system.

By way of illustration, therefore, I will illustrate the use of the framework by reference to an outline of the model's application to the development of Western art music during the Second Millennium AD.

I will conclude by suggesting that, once formalised, the uniformity and power of Boden's proposal becomes rather more clear than before, and that it may not be idly dismissed as vague, as it sometimes has been in the past.

2.2 Background: Boden's Analysis of Creative Systems

Boden (2004) aims to study the idea of AI-based simulation of creativity from a philosophical viewpoint. She begins by setting out two taxonomies of creative behaviour, in two orthogonal ways.

First, she makes the distinction between H- and P-creativity: creativity which is 'historical' or 'psychological', respectively, the latter being interchangeable with 'personal', should that be more natural for the reader. The distinction is, respectively, between the sense of creating a concept which has never been created before – ever, anywhere – and a concept which has never been created before by a specific creator. This distinction will be only tangentially relevant to my argument here, but before proceeding, I note that this simple binary characterisation is over-simple: it would be possible, for example, for a creative behaviour to be only P-creative in one society, but H-creative in another; from the point of view of the second society, only the H-

creativity matters. Wiggins, Tyack, Scharff, and Rohrmeier (2015) propose that this question of *novelty in context* should be modelled as a four-place relation, between artefact/concept, creator, context and observer.

Second, in Boden's work, there is the distinction between *exploratory* and *transformational* creativity, which is directly relevant here, and so needs a little more explanation. Boden conceives the process of creativity as the identification and/or location of new conceptual objects in a *conceptual space*. Subsequent authors have sometimes imagined the conceptual space to be the state space of 'Good Old-Fashioned AI' (Russell & Norvig, 1995), though it is not clear that Boden intends her proposal to be taken so specifically or literally; Wiggins (2006b) discusses this question in more detail.

If we do go along with Boden's conception, for the sake of argument, then the creative act might be said to be exploring a space of partial and complete possibilities, and this is the kind of creativity which Boden calls *exploratory*. The existence of such a conceptual space begs a question (at least to the AI researcher): what rules define the space? If there are rules which define the space, then, presumably, those rules can be changed, producing what might be thought of as a paradigm shift. This kind of change is *transformational creativity* to Boden.

However, it is not clear from Boden's writings about these ideas how she would define the constraints which give rise to a particular conceptual space, and therefore what is the difference, in terms of the new concepts discovered, between exploring the space and transforming it. I will argue below that there is at least one way we can coherently make such a distinction. First, however, it is necessary to sharpen slightly the philosphical tools that Boden introduced.

2.3 Terminology

Some 40-odd years ago the discipline of artificial intelligence was identified and named. One issue that dogged it then, and still does now is 'what – exactly – is the intelligence it claims to create and/or simulate and/or emulate?' Many attempts at answers have been made, some more successful than others, some evidently confusing intelligence with consciousness. One definition that seems to have some hope of enduring circumvents the problem of saying what intelligence *is* by restricting its definition to where it *resides*:

> The performance of tasks which, if performed by a human, would be deemed to require intelligence.[1]

This definition is in many ways unsatisfactory: for example, it does not include many of the fundamental aspects of robotics and machine vision which we do normally include in the field of AI. However, it will do nicely for the current purpose,

[1] This definition is now part of the AI folklore, being attributed to various authors, including Minsky and Turing. Its true original source is obscure to me; I apologise, therefore, to the original author for my lack of citation.

because it does capture the parts of AI which are concerned with higher cognitive function, such as mathematical reasoning, construction of language semantics, and artistic pursuits, including painting and music.

The zenith of human intelligence is very often portrayed as the ability to create, and to create radically new and/or surprising things. For example, in humans, the acts of elaborating a new and elegant mathematical proof, or designing a subtle new experiment to investigate the existence of a new subatomic particle (the prediction of which is itself a creative contribution), or writing a novel, painting a watercolour, or composing a sonata are all deemed the height of creativity, and therefore the height of (some aspects of) intelligence.

Throughout human society, creative individuals and groups are valued very highly, and creative behaviour is fundamental in that society. For example, musical behaviour is a uniquely human trait (notwithstanding our anthropomorphic tendencies in terminology such as bird- and whale-'song', which are in fact much more like language than music); further, it is also ubiquitously human: there is no known human society which does not exhibit musical behaviour in some form.[2]

It seems, then, that at least commonplace definitions of creativity place it as a primary determiner of human intelligence. As such, we might expect it to be at the forefront of AI research, but it is not – at least, not explicitly. Boden (1977) raised the issue for what seems to be the first time in AI literature, but was apparently rebuffed:

> . . . when (in the early 1970s) I included a chapter on creativity in my book Artificial Intelligence and Natural Man [Boden (1977, Chapter 11)], most people – including my AI-friends – asked me in puzzlement 'Why on earth are you doing that?'
>
> (Boden, 1999, p. 11)

Perhaps creativity is to AI researchers what intelligence seems to be to some of those computer scientists and philosophers who continue to argue against AI: that feature which is best left alone, lest we cease to be distinguishable from machines, and belief in whose attainment is viscerally beyond the pale.

Returning, then, to the topic of the present section: how can we define 'creativity' and thence 'computational creativity'? It turns out that one good way to do so is to adapt the definition of intelligence I gave above. This is a good strategy for two reasons: like 'intelligence', 'creativity' is ill-defined, but we do tend to know it when we see it; worse than 'intelligence', 'creativity' as a word is overloaded, and is usable in distinctly different and *confusing* ways. First, then, I suggest that a useful working definition of 'computational creativity' is

> The performance of tasks which, if performed by a human, would be deemed creative.

Note that this definition includes the production of creative artefacts, which may be deemed 'more' or 'less creative' in some usages of the C-word.[3]

---

[2] For an unscientific thought-demonstration of music's ubiquity, the reader may wish to try to name a Western social or ceremonial occasion in which music is not usually involved at some level. The legal courts are the only example this author has found.

[3] The question, 'what is creativity?' has also been addressed at length by Colton and Wiggins (2012).

From this, my personal definition of 'computational creativity' is

> The study and support, through computational means and methods, of behaviour exhibited by natural and artificial systems which would be deemed creative if exhibited by humans.

This 'study and support' may, of course, include simulation. Having now given a definition, however intangible, of the word 'creativity', I will now forswear most of its diverse usage, on the grounds that it is already defined in too many ways to carry yet another meaning, no matter how precise. I shall use the following terminology:

Creative system. A collection of processes, natural and/or technological, which are capable of achieving or simulating behaviour which, if exhibited by humans, would be deemed creative.

Creative behaviour. One or more of the behaviours exhibited by a *creative system*.

Novelty. The property of an artefact (abstract or concrete) output by a *creative system* which arises from prior non-existence of like or identical artefacts in the context in which the artefact is produced.

Value. The property of an artefact (abstract or concrete) output by a *creative system* which renders it desirable in the context in which it is produced.

Armed with these terms, I will attempt to avoid use of the word 'creativity' itself altogether, and thus avoid the imprecision and ambiguity it invariably entails. The exception to this will be in the use of terminology introduced by other authors.

It is perhaps worth mentioning some other particular terminological issues at this point. Some authors, such as Macedo and Cardoso (2001), seem to view 'surprise' as a property of a creative system. I would argue that 'surprise' is a property of the receiver of a creative output: it is an emotion generated by either the novelty of the output or (cynically) by the unexpected ability of the creative system to produce something of value. It is also common in everyday language to hear an artefact being called 'creative' (for example, 'That's a creative painting!'). What this usually means is that the perceiver finds a blend of novelty and value in the artefact; it needs to be distinguished from the claim that the artefact is itself a creative system, which could be expressed by the same sentence.

Having pinned down the terminology a little, I now proceed to discuss my formalisation of Boden's informal framework. I refer to the result as the creative systems framework (CSF).

2.4 A Framework for the Description of Creative Systems

2.4.1 A Universe of Possibilities

Boden's combination of the idea of the conceptual space with distinct notions of exploratory and transformational creativity has some consequences which are left implicit in her published work. Most fundamentally, for transformational creativity to have any meaning, there must be a universe of possibilities, which I shall call

$\mathscr{U}$, which is a *non-strict superset* of the conceptual space at any given point in the creative process. To see the reason for this, let us first define $\mathscr{U}$.

Definition 2.1 (Universe). The *universe*, $\mathscr{U}$, is a multidimensional space, whose dimensions are capable of representing anything, and all possible distinct concepts correspond with distinct points in $\mathscr{U}$.

For parsimony, we could restrict $\mathscr{U}$ to be capable of representing just the things which are relevant to the domain in which we wish to be creative – but this would rule out cross-domain transfer of concepts, by processes such as analogy, which would be undesirable in general. (I return to analogy and other means of guiding exploratory creativity below.) To make the proposal as state-space-like as possible, I allow that $\mathscr{U}$ contains all abstract concepts as well as all concrete ones, and that it is therefore possible to represent both complete and incomplete artefacts. Henceforward I will refer to both incomplete and complete concepts simply as 'concepts', except where the distinction is significant. It follows from the inclusion of incomplete concepts that we should also admit the most incomplete concept of all, the empty concept, which I will denote by $\top$, and that it should be a member of $\mathscr{U}$.

To summarise, the following points are axiomatic to my formulation. These axioms cannot be stated within the formulation, because they describe its own properties, and not just those of the system it models.

Axiom 1 (Universality) *All possible concepts, including the empty concept,* $\top$*, are represented in* $\mathscr{U}$*; thus,* $\mathscr{U}$ *is the type of all possible concepts.* $\top \in \mathscr{U}$.

Axiom 2 (Non-identity of concepts) *All concepts* c_i *represented in* $\mathscr{U}$ *are mutually non-identical.* $\forall c_1, c_2 \in \mathscr{U}.c_1 \neq c_2$.

We need $\mathscr{U}$ because, if a conceptual space were *equal to* $\mathscr{U}$ (and $\mathscr{U}$ were therefore superfluous), any point in $\mathscr{U}$ could be reached by exploration. Therefore, transformation would be meaningless. So, for transformational creativity to be meaningful, all conceptual spaces $\mathscr{C}$ are required to be strict subsets of $\mathscr{U}$.

Axiom 3 (Universal inclusion 1) *All conceptual spaces* $\mathscr{C}$ *are strict subsets of* $\mathscr{U}$. $\forall \mathscr{C}.\mathscr{C} \subset \mathscr{U}$.

Axiom 4 (Universal inclusion 2) *All conceptual spaces* $\mathscr{C}$ *include* $\top$. $\forall \mathscr{C}.\top \in \mathscr{C}$

So far, I have done nothing more precise than Boden's informal characterisation; I have merely pinned the ideas down to a specific formulation and pointed out a logical consequence (the necessity for the existence of $\mathscr{U}$ as distinct from $\mathscr{C}$). It is in the definition of $\mathscr{C}$, in terms of its own constraints, rather than its relation to $\mathscr{U}$, that we first find the necessity to clarify the existing ideas.

2.4.2 Defining the Conceptual Space

Boden (2004) does not explicitly acknowledge the existence of (an equivalent of) $\mathscr{U}$. Instead, she loosely defines her conceptual spaces in terms of a set of definitional

rules, which we must therefore assume to be generative. However, she blurs the distinction between the rules which determine membership of the space (i.e., which select members of $\mathscr{U}$ to be members of a particular $\mathscr{C}$, in my terms), and other rules which might allow the construction and/or detection of a concept represented by a point in the space. To remedy this, let us take two distinct rule sets, $\mathscr{R}$ and $\mathscr{T}$, being rules which constrain the space and rules which allow us to traverse it, respectively. In AI terms, then, $\mathscr{T}$ might be thought of as encoding a search strategy, perhaps including heuristics.

In order to introduce these sets of rules, we need a language in which to express them; it will be convenient to arrange that both sets can be expressed in the same language.[4] I will call this language $\mathscr{L}$, and formalise it as the set of all sequences composed of some alphabet, $\mathscr{A}$. Therefore, by definition,

$$\mathscr{R} \subset \mathscr{L}, \ \mathscr{T} \subset \mathscr{L}.$$

Given the language, $\mathscr{L}$, and rule sets expressed in it, we need an interpretation function, $[\![.]\!]$, which is a partial function[5] from $\mathscr{L}$ to functions yielding real numbers in $[0, 1]$. At this point, we will use a real value greater than or equal to 0.5 to mean Boolean `true` and less than 0.5 to mean Boolean `false`; later, we will allow ourselves the option of considering partial membership of conceptual spaces (as in fuzzy set theory) or probabilistic uncertain reasoning, either of which requires non-binary quantities. This Boolean decision procedure will allow us to choose the members of $\mathscr{U}$ we want in $\mathscr{C}$, assuming that $\mathscr{R}$ is well-formed with respect to $[\![.]\!]$:

$$\mathscr{C} = [\![\mathscr{R}]\!](\mathscr{U}).$$

2.4.3 Exploring the Conceptual Space

A similar approach is required for the application of the search strategy encoded in $\mathscr{T}$, though a little more computational mechanism is required. We need a means not just of *defining* the conceptual space, irrespective of order, but also, at least notionally, of *enumerating* it, in a particular order, under the control of $\mathscr{T}$ – this is crucial to the simulation of a particular creative behaviour by a particular $\mathscr{T}$. By analogy, again, with the familiar approach to state space search, I introduce a further interpreter $\langle\langle .,.,. \rangle\rangle$, which, given three well-formed subsets of $\mathscr{L}$, computes a function which maps c_{in}, a sequence of elements[6] of $\mathscr{U}$, to c_{out}, another sequence of elements of $\mathscr{U}$. As with $\mathscr{R}$ and $[\![.]\!]$, I assume that $\mathscr{T}$ does not contain subsequences which have

[4] This is always possible: a language can always be the union of two others, given any necessary renaming apart.

[5] Partial, because I have not required that $\mathscr{L}$ contain only sequences which are well-formed with respect to $[\![.]\!]$. The reason for this will become clear below.

[6] These sequences fulfil the role of the agenda, before and after expansion, where the CSF is used to model a traditional search mechanism (Wiggins, 2006b).

no interpretation under $\langle\langle .,.,. \rangle\rangle$. The three arguments of $\langle\langle .,.,. \rangle\rangle$ are, respectively, the rule set defining the conceptual space, $\mathscr{R}$, the traversal strategy, $\mathscr{T}$, which the function is intended to apply, and another set, $\mathscr{E}$, which I will define below; this gives $\mathscr{T}$ the possibility of being informed by $\mathscr{R}$ and $\mathscr{E}$. The resulting function operates on members of $\mathscr{U}$ and not just on members of $\mathscr{C}$, as one might expect, because it is necessary to describe and simulate behaviours which are not merely exploratory. More on this below.

The ordering on c_{in} and c_{out} indicates the order in which the concepts in them are to be next considered for further development under $\mathscr{T}$, so the input and output of the function are successive states of a kind of agenda:

$$c_{\text{out}} = \langle\langle \mathscr{R}, \mathscr{T}, \mathscr{E} \rangle\rangle (c_{\text{in}}).$$

However, note that this formulation is more powerful than the standard formulation of AI state space search because the function is allowed to select arbitrarily many members of c_{in} and is not limited to the head(s) of the sequence. This is a key feature, because it admits search strategies which rely on the combination of or comparison between agenda items.

It follows that we would begin some of our creative processes by computing

$$\langle\langle \mathscr{R}, \mathscr{T}, \mathscr{E} \rangle\rangle (\{\top\}).$$

We now have all the mechanism we need to model Boden's exploratory creativity as presented in 2004.

2.4.4 The Value of Two Rule Sets, $\mathscr{R}$ and $\mathscr{T}$

Importantly, separating the rules out into the sets $\mathscr{R}$ and $\mathscr{T}$ has given us the ability to consider alternative versions of $\mathscr{T}$ with any given $\mathscr{R}$, and, perhaps less obviously, vice versa. We can partition $\mathscr{C}$ into $\mathscr{C}_!$, concepts which have already been discovered, and $\mathscr{C}_?$, concepts which have not. Some versions of $\mathscr{T}$ may be effective in traversing $\mathscr{C}$ and in finding members of $\mathscr{C}_?$; some may be less so; and some may be good at finding members of $\mathscr{C}_?$ in some parts of $\mathscr{C}$ and not in others. Further, some elements in $\mathscr{C}$ may not even be accessible under $\mathscr{T}$. So now we have, for example, the ability to simulate two composers working in the different ways within the same style, for example, which was less clear in Boden's simpler formulation.

It is worth noting, also, why $\mathscr{T}$ does not supplant $\mathscr{R}$ as a primary component of the framework: this is because, in any real simulation of creativity, there is a distinction between something being an X (where X is the kind of thing to be created) and it being a *valued* X (Ritchie, 2007). $\mathscr{E}$ defines the value; $\mathscr{R}$ defines the nature of the created artefact. In particular, in the societal context, $\mathscr{R}$ represents the agreed nature of what the artefact is, in the abstract; this is distinct from $\mathscr{T}$, which defines the way a particular agent produces an artefact in practical terms.

2.4.5 Evaluating Members of the Conceptual Space

To do full justice to Boden's model as presented in 2004, we need one further set of rules, $\mathscr{E}$, such that $\mathscr{E} \in \mathscr{L}$. This is the set of rules which allows us to evaluate any concept we find in $\mathscr{C}$ and determine its quality, according to whatever criteria we may consider appropriate – and, of course, it is not hard to imagine that $\mathscr{T}$ might be related to or dependent on $\mathscr{E}$, which is why the interpretation function includes $\mathscr{E}$ in its parameters. However, I am making no attempt here to discuss or assess the value of any concepts discovered: while this issue is clearly fundamentally important (Boden, 1998; Pearce & Wiggins, 2001; Ritchie, 2007, 2019), it can safely be left for another time. Ritchie (2019) gives a detailed approach to the evaluation of creative systems as such, and it is self-evident that some of the ideas presented might be used in $\mathscr{E}$. Suffice it to say here that the existing function $[\![.]\!]$ will be adequate to select those results of the creative process which are 'valued' by $\mathscr{E}$, thus:

$$[\![\mathscr{E}]\!](\langle\langle \mathscr{R}, \mathscr{T}, \mathscr{E}\rangle\rangle^{\diamond}(\{\top\})),$$

where

$$\mathscr{F}^{\diamond}(X) = \bigcup_{n=0}^{\infty} \mathscr{F}^{n}(X),$$

$\mathscr{F}$ being a set-valued function of sets. As before, a more or less nuanced view can be implemented using either a real value in $(0,1)$ or a Boolean computed by thresholding the continuous value.

2.4.6 Characterising an Exploratory Creative System

To summarise, we now have the machinery to describe an exploratory creative system in these terms with the following septuple:

$$\langle \mathscr{U}, \mathscr{L}, [\![.]\!], \langle\langle .,.,.\rangle\rangle, \mathscr{R}, \mathscr{T}, \mathscr{E}\rangle.$$

Many of the systems described in this anthology may be described within this framework, with many of the technical differences being located in $\mathscr{L}$or, specifically, the part of $\mathscr{L}$ that defines the knowledge representation, or in $\mathscr{T}$, the traversal mechanism. Some systems, such as MEXICA (Pérez y Pérez, 2019), which are exploratory, have a superficial appearance of transformation due to their reflection on their productions, which must nevertheless be properly understood as traversal, $\mathscr{T}$, notwithstanding the fact that it appeals to rules of value, $\mathscr{E}$, in its process. However, a creative system that is limited to exploratory behaviour is unlikely to create something that changes the world, though this, *pacet* Boden (2004), is not impossible (Wiggins, 2006b).

2.4.7 Exploring and Transforming

Before proceeding to the formality of transformational creativity, there are some more issues to discuss in the exploratory context.

It follows from my characterisation of $\mathscr{T}$ as a search engine that there may be a fitness hypersurface associated with any combination of $\mathscr{C}$ and $\mathscr{T}$. The 'landscape' so defined may be arbitrarily – perhaps extremely – convoluted. This means that it is possible to imagine finding c, a member of $\mathscr{C}_?$, which is, in general, very hard to find, but doing so *without changing* $\mathscr{T}$. Finding such a concept would presumably mark the creator as successful, especially if other creators' $\mathscr{T}$s were less fortunate, for the discovery is unlikely. So here is a case where an exploratory creation might well be very significant – perhaps more significant than many transformational creations.

Now, consider the converse situation. Suppose that a concept c is a member of $\mathscr{U}$, but not a member of $\mathscr{C}$, and that we *transform* $\mathscr{C}$ into $\mathscr{C}'$ by transforming $\mathscr{R}$ into $\mathscr{R}'$ – I discuss how this can happen below. Now we have exhibited transformational creativity, which, according to Boden, is more significant than exploratory creativity. But it may be the case that

$$\mathscr{C}' = \mathscr{C} \cup \{c\},$$

in which case it is hard to argue that the transformation is any more significant than the exploration in the account immediately above, unless, of course, c is very significant in itself.

Now let us consider a third possibility, one which was not available to Boden because of her conflation of my $\mathscr{R}$ and $\mathscr{T}$: it is possible in principle for a concept which exists in $\mathscr{C}$ – and so is sanctioned by the constraints in $\mathscr{R}$ – to be unreachable by the rules specified in $\mathscr{T}$. This is an important point: it distinguishes what is *in principle* possible in a creative domain from what is *actually* possible according to the properties of a given creator. Therefore, another possibility for reaching the elusive discovery c above is that

$$c \in \mathscr{C},$$

but the rules of $\mathscr{T}$ fail to make it accessible:

$$c \notin \langle\langle \mathscr{R}, \mathscr{T}, \mathscr{E} \rangle\rangle^{\diamond}(\{\top\}).$$

So we have to introduce a new, different notion of transformational creativity – one which transforms not $\mathscr{R}$, the rules constraining the conceptual space, but $\mathscr{T}$. It is not hard to imagine that we can transform $\mathscr{T}$ into some $\mathscr{T}'$ such that c becomes accessible to our search.

From an external viewpoint, these different events are probably often indistinguishable, but the point is that they all fall short of Boden's informal definition of transformational creativity (that is, in the terms used here, changing $\mathscr{R}$) – which she argues is generally more significant than the exploratory kind.

So by making the argument more precise, we can demonstrate a potential weakness in Boden's characterisation: the boundary between exploratory and transformational

creativity is ill-defined.[7] We are now in a position to argue that transformational creativity is unnecessary, and to conflate $\mathscr{U}$ and $\mathscr{C}$, thus producing a simpler characterisation.

However, I will argue that there is indeed a valid distinction between a kind of creative behaviour that might be called 'transformational' and a kind of creative behaviour that might be called 'exploratory'. Before I can do so, however, we must consider transformational creativity in more detail.

2.4.8 Transformational Creativity

Having gone some way towards formalising Boden's notion of exploratory creativity, we are now in a better position to say what transformational creativity actually is. It is at this point that we begin to see the benefits of this laborious formalisation. In this section, I discuss transformational creativity informally; I will treat it more formally in a later section.

Boden characterises her 'transformational creativity' as the kind of creative behaviour concerned not with finding members of $\mathscr{C}_?$ in a given conceptual space $\mathscr{C}$, but with transforming the rule set defining $\mathscr{C}$ so as to produce a new conceptual space, $\mathscr{C}'$. In my terms, this might be achieved in two essential ways: by transforming $\mathscr{R}$ or by transforming $\mathscr{T}$ (recall that, although changing $\mathscr{T}$ does not, by definition, change $\mathscr{C}$, any given $\mathscr{T}$ cannot guarantee to reach all the elements of $\mathscr{C}$ – so a new $\mathscr{T}'$ may make a different subset of $\mathscr{C}$ *available*). Transforming $\mathscr{R}$ corresponds with changing the rules of the creative game being played – and, it seems, with what Boden calls 'transformational' creativity. The new, second kind of transformation, of $\mathscr{T}$, more naturally applies to the creative individual's *modus operandi* only – there seems to be no explicit analogue of this in Boden's formulation. Of course, it is possible for both kinds of transformation to happen at once. To distinguish between these two different kinds of transformation, I will use the terms $\mathscr{R}$*-transformational* and $\mathscr{T}$*-transformational*; however, in using Boden's terminology, I shall continue to use the unadorned 'transformational'.

Now, as mentioned earlier, Boden concludes, from what she portrays as historical precedent, that her transformational creativity (i.e. $\mathscr{R}$-transformational) is somehow more significant than exploratory creativity. This claim deserves some more scrutiny in the light of my division of the creative rules into $\mathscr{R}$ and $\mathscr{T}$. Let us consider the difference between these two.

Suppose, as Boden supposes, that $\mathscr{R}$ defines a set of concepts which is largely agreed among all creative agents interested in the area defined by $\mathscr{R}$. Then, almost by definition, any change in $\mathscr{R}$ has the force of a paradigm shift (even if only a little one), if it is valued highly enough by the *existing* evaluation rule set, $\mathscr{E}$, because it changes the *agreed* rules of the game. To ground this in an example, consider Kekulé's discovery of benzene rings, cited repeatedly by Boden (2004) as

[7] This author is by no means the first to note this point.

an example of transformational creativity. The idea was new because it allowed *rings* of carbon atoms, and not just chains. But the evaluation system was independent of the shape used: it was a meta-theoretic evaluation of whether the theory explained the chemical data. Thus, Kekulé's new rule set was valued more highly under the *existing* evaluation rules than were the pre-existing solutions. In this volume, a similar kind of $\mathscr{R}$-transformation is proposed by Correia, Machado, Romero, Martins, and Cardoso (2019).

On the other hand, $\mathscr{T}$, as I have proposed it, is not global or agreed: it is the technique of the individual creator. Therefore, a change in $\mathscr{T}$ is on a different scale from a change in $\mathscr{R}$: it may perhaps accelerate the agent's progress towards a good solution; it may even make accessible concepts which were not previously available to this particular agent, but it will not change the nature of the space of possibilities, and therefore will not constitute a paradigm shift. An archetype here would be the comparison between an expert organist, capable of convincingly harmonising a chorale melody at first sight in the style of J. S. Bach, and a beginning harmony student, struggling to do so for the first time. The rules of Bach chorale harmony ($\mathscr{R}$) may be common to both, but the techniques ($\mathscr{T}$) of the two are not.

For completeness, it is necessary to consider the case where an $\mathscr{R}$-transformation is not necessarily adopted by all the creative agents working on $\mathscr{R}$. This case has, of course, been seen many times in history.[8] It can arise reasonably only where different creative agents working in an initially common $\mathscr{R}$ have different evaluation rule sets, $\mathscr{E}_i$ – the alternative case, where there are initially differing $\mathscr{R}_i$s, does not correctly describe the example situation. This raises an interesting question of how discovery of new ideas can lead to changes in the evaluation rule set itself; this will be a focus of future work.

2.4.9 Creative Behaviour and the Meta-level

An aspect of this discussion which Boden (2004) leaves implicit is the formal relationship between exploratory and transformational creativity – one would need a formalisation of the kind presented here to do so. I now extend that formalisation to cover transformational creativity, and will show that, as informally conjectured by Bundy (1994), we can view transformational creativity as exploratory creativity at the meta-level, and thus keep the framework simple.[9]

The idea at the root of Boden's ($\mathscr{R}$-)transformational creativity is that of changing the rules which define her conceptual space. In my formulation, there are two such rule sets, $\mathscr{R}$ and $\mathscr{T}$. So, in my terms, transformational creativity consists in changing either $\mathscr{R}$ or $\mathscr{T}$ or both. The two sets are expressed in the language $\mathscr{L}$, which means that the result of the transformation(s) must also be in $\mathscr{L}$. We can place a useful restriction on the results of these transformations: that they be well-formed in terms

[8] The continuing prevalence of tonal harmony in music long after the arrival of modernism is one example.

[9] Buchanan (2001) argues that creative behaviour arises only at the meta-level.

of whatever interpreter will interpret them. So we need a syntax checker which will select the elements of $\mathscr{L}$ which are well-formed in that sense.

Transformation of either kind means constructing new subsets of $\mathscr{L}$ from old ones. Starting from the empty sequence, or from anywhere else in the space of possibilities, we can do this by application of a search algorithm. If we allow ourselves access to a meta-language $\mathscr{L}_{\mathscr{L}}$ for $\mathscr{L}$, which can describe the construction of new members of $\mathscr{L}$ from old ones, we can pair it with an appropriate interpreter, to allow us to search the space of possibilities. Since the syntax-checking task mentioned above is a meta-level structural one (with respect to $\mathscr{L}$), we can use $\mathscr{L}_{\mathscr{L}}$ to describe this task too, again given an appropriate interpreter. Finally, we need to be able to evaluate the quality of the transformational creativity, with some function Ω.

By now, the reader will see where this argument is going. We can specify interpreters, $[\![.]\!]$ and $\langle\langle .,.,. \rangle\rangle$, which will interpret a rule set $\mathscr{T}_{\mathscr{L}}$ applied to an agenda of potential sequences in $\mathscr{L}$; if we are careful, we can specify such an interpreter which will work for both $\mathscr{L}$ and $\mathscr{L}_{\mathscr{L}}$. Finally, we can express our evaluation function, Ω, as a set of sequences, $\mathscr{E}_{\mathscr{L}}$, in $\mathscr{L}_{\mathscr{L}}$, and use $[\![.]\!]$, to execute it. This time, we can allow the real-valued output of the interpreter to be used either for comparison or (as before) for selection, depending on context. Our transformational creativity system can now be expressed as

$$\langle \mathscr{L}, \mathscr{L}_{\mathscr{L}}, [\![.]\!], \langle\langle .,.,. \rangle\rangle, \mathscr{R}_{\mathscr{L}}, \mathscr{T}_{\mathscr{L}}, \mathscr{E}_{\mathscr{L}} \rangle,$$

or, in other words, as an exploratorily creative system working at the meta-level of representation.

The only connection we have now not considered is that (if any) between $\mathscr{E}$ and $\mathscr{E}_{\mathscr{L}}$. I suggested above that, for a transformationally creative act to be valued, it would normally need to be valued under the criteria that governed the original search space. This begs the question of how to relate $\mathscr{E}$, which is defined over the exploratory/object-level universe $\mathscr{U}$, to the transformational/meta-level universe, $\mathscr{L}$.

Informally, and minimally, the transformation is valued if it admits a new concept which is valued to the available object-level conceptual space. We can express this, in terms of the exploratory creative system described above, by saying that $\mathscr{E}_{\mathscr{L}}$ is the rule set which selects pairs of $\mathscr{R}_{\mathscr{L}}$ and $\mathscr{T}_{\mathscr{L}}$, such that

$$\{c \mid r \in \langle\langle \mathscr{R}_{\mathscr{L}}, \mathscr{T}_{\mathscr{L}}, \mathscr{E}_{\mathscr{L}} \rangle\rangle^{\diamond}(\{\mathscr{R}\}) \wedge t \in \langle\langle \mathscr{R}_{\mathscr{L}}, \mathscr{T}_{\mathscr{L}}, \mathscr{E}_{\mathscr{L}} \rangle\rangle^{\diamond}(\{\mathscr{T}\}) \wedge c \in [\![\mathscr{E}]\!]\langle\langle r,t,\mathscr{E} \rangle\rangle^{\diamond}(\{\top\})\} \neq \emptyset.$$

In other words, $\mathscr{E}_{\mathscr{L}}$ is the rule set which selects pairs of $\mathscr{R}_{\mathscr{L}}$ and $\mathscr{T}_{\mathscr{L}}$ such that new concepts are added to the conceptual space under consideration, and that those new concepts are valued by $\mathscr{E}$.

This meta/object-level distinction raises some interesting questions. The most obvious is: if this relation holds between the object level of our creative domain and the meta-level of transformational creativity, what would it mean to take the same relation and apply it to the transformational level? However, I will leave these issues for future work.

To conclude the current section, let us return to the issue of the relative values of exploratory and transformational creativity, as introduced by Boden (2004). I have

argued above that Boden's suggestion that transformational creativity is innately superior to exploratory creativity is not well founded in terms purely of the creative product. However, the meta-level notion of transformational creativity which I have introduced above gives us another means of looking at the question, a means which Boden does not (at least, explicitly) use.

I suggest that, for true transformational creativity to take place, as described in my framework above, the creator needs to be in some sense *aware* of the rules he/she/it is applying. This follows from the need to explore the space of possible rule sets defining the conceptual space. One might argue that serendipity – a happy accident – might account for creative behaviour, and this can certainly be the case, but that would be a new category, of 'serendipitous' creativity, and not transformational creativity, under either of my definitions. I make this point because it fits in very clearly with philosophical notions of art. Reflection is generally cited as the property which distinguishes the artist from the craftsperson.[10] That self-awareness, I suggest, is what makes a creator able to formalise his/her/its own $\mathscr{R}_{\mathscr{L}}$ and $\mathscr{T}_{\mathscr{L}}$ in terms of the meta-language $\mathscr{L}_{\mathscr{L}}$. So, without that self-awareness, a creator *cannot* exhibit transformational creativity, though, conversely, of course, a creator *with* self-awareness may choose not to exercise it.

2.4.10 Combinatorial Creativity

Boden (2004) introduces a third kind of creativity: combinatorial creativity (which has sometimes been misquoted as combinational creativity). The idea here is to capture the creation of new concepts or artefacts which are made by the combination of features of other existing concepts or artefacts, in particular from different domains, in an operation presumably similar to the *bisociation* operation of Koestler (1976), which is discussed in detail later in this volume (Lavrač et al., 2019).

The key point about combinatorial creativity is that of bringing together components from concepts in two different conceptual spaces, and combining them into something new, possibly in a third, different conceptual space.

To express this idea in the proposed framework, one needs either two conceptual spaces, defined by two different rule sets, or one notional conceptual space which is defined as the union of the two distinct conceptual spaces. Either approach may be straightforwardly represented, because whatever happens must be formulated in the context of a given $\mathscr{U}$, which means that all the necessary dimensions are already available for the representation, and any new concept (and possibly conceptual space) generated by the operation can also be located within $\mathscr{U}$.

[10] Whether this is a valid distinction is an orthogonal issue, not discussed here.

2.4.11 Creative Behaviour Is Not Just Traditional AI Search

One reason that Boden's work on creativity is criticised by some AI researchers is that exploratory creativity, on the surface, seems to be very much like the familiar state space search in the AI literature, sometimes known as 'Good Old-Fashioned AI'. I suggest, however, that the characterisation above highlights significant differences. These differences have recently been identified by some researchers as the distinction between AI as practised, based on problem-solving, and artificial general intelligence (AGI), an altogether more meta-level activity, in which the solutions to problems are assumed unknown. This distinction changes the fundamental nature of search-based formulations, because one cannot choose one's representation to suit one's solution.

First, in the work on creative systems, there is no function which can definitively detect a 'solution'. Indeed, there is not necessarily even a clear 'problem' to which a 'solution' is sought: the dimensions defining the conceptual space may simply describe artefacts which have no (truth-functional) meaning at all.

Second, there is a technical difference: in standard state space search, we normally operate on one node (which may indeed represent a partial solution) at a time. In my formalisation, at least, traversal of the space may arise through simultaneous consideration of (and hence consideration of the relationship *between*) more than one of the nodes in it which have already been discovered. Thus the search pattern produced is not a tree, but a lattice. (Of course, more modern search methods, such as genetic algorithms, address this issue directly.)

Third, also on the technical level, the distinctions between the boundary of the conceptual space ($\mathscr{R}$), and the evaluation scheme ($\mathscr{E}$) are not explicit in standard search algorithms, unless $\mathscr{R}$ is to be found in the syntax of the representation, and $\mathscr{E}$ is the stopping criterion, which are relatively blunt instruments. These latter possibilities are at odds with at least my formulation of creative behaviour, because, as we will see below, a creative agent needs to be able to create outside the acceptability of $\mathscr{R}$, and because $\mathscr{E}$ is by no means an ending detector, but an indication of the quality of a solution (and not in the same sense as a heuristic, because it is not primarily used to guide the search).

Fourth, and most importantly, this is a multilayered system (though I have addressed only two layers here). The application of transformational creativity means that the creative agent is a self-organising, self-evaluating system, and this is far beyond the scope of traditional AI search.

However, it is clearly the case that one could implement AI search in the exploratory framework I propose above. Therefore, I suggest that standard AI search is in fact a special case of a creative system, and that it is not unreasonable to claim that at least those search-based systems which use heuristics are exhibiting creative behaviour, under the definition I gave above.

2.5 Generic Application of the Framework

2.5.1 Introduction

A theoretical framework, such as that presented here, may be useful in teasing out philosophical issues, but it may also be useful in giving generalised descriptions of behaviours which might be observed in creative agents. In this section, I show how to use the tools provided in earlier sections to identify certain potentially important behaviours of creative systems, and suggest ways in which they might be addressed. It is to be emphasised that the aim here is not to ground this particular framework as an implementation, but to show how it might be used to describe implementations and some of their properties in useful ways.

2.5.2 Useful Properties of Creative Agents

The apparent supposition in Boden's work is that creative agents will be well behaved, in the sense that they will either stick within their conceptual space or alter it politely and deliberately by transformation. It can be argued, however, that this is not adequate to describe the behaviour of real creative systems, natural or artificial, either in isolation or in a societal context. This section identifies some situations not covered by the assumption of good behaviour, and gives names to them. The important point is that some of these situations may appropriately trigger particular events, such as a step of transformational creativity, so it is useful to be able to identify them in the abstract. This leaves us with several general classes of small-scale conditions which might be observed in AI systems, of which we can then assess the creative potential.

2.5.2.1 Uninspiration

There are various ways that a creative system can fail to be creative in a valued way. These ways can be characterised through the rule set $\mathscr{E}$. Alongside each diagnosis, in the list below, I list potential treatments.

Hopeless uninspiration is the simplest case, where there are no valued concepts in the universe:

$$[\![\mathscr{E}]\!](\mathscr{U}) = \emptyset.$$

This system is incapable, by definition, of creating valued concepts, and as such might be termed ill-formed (if such creative behaviour is the intention). Therefore, it is up to the system designer to remedy the problem, like a *deus ex machina*.

Conceptual uninspiration arises when there are no valued concepts in the conceptual space:

$$[\![\mathscr{E}]\!]([\![\mathscr{R}]\!](\mathscr{U})) = \emptyset.$$

I label this form of uninspiration 'conceptual' because it entails a mismatch between $\mathscr{R}$ (which defines the conceptual space) and $\mathscr{E}$ (which evaluates concepts within it, and, more broadly, within $\mathscr{U}$). This condition is contradictory to the purpose of the two rule sets: if $\mathscr{R}$ is supposed to constrain the domain of a creative process, then it is inappropriate for $\mathscr{E}$ not to select some of the elements it admits. As such, like the hopeless case, conceptual uninspiration indicates ill-formation of the intended creative system.

Conceptual uninspiration can be remedied by transforming $\mathscr{R}$, or by modifying $\mathscr{E}$. How $\mathscr{R}$ and/or $\mathscr{E}$ should be modified is an open question, but *aberration*, defined below, can act as a trigger and possibly a guide to what to transform.

Generative uninspiration occurs when the technique of the creative agent does not allow it to find valued concepts within the space constrained by $\mathscr{R}$:

$$[\![\mathscr{E}]\!](\langle\langle \mathscr{R}, \mathscr{T}, \mathscr{E}\rangle\rangle^{\diamond}(\{\top\})) = \emptyset.$$

This kind of uninspiration is less problematic than the other two, and does not necessarily indicate an ill-formed creative system: it merely indicates that a creative agent is looking in the wrong place. This raises the question of *why* there is such a mismatch. Boden's underlying assumption seems to be that the conceptual space is in some sense definitive, and, certainly, in a multi-agent environment, it is the only place in the formalism where the consensus about a creative domain can logically be represented. Therefore, I propose that the usual solution to generative uninspiration should be transformation of $\mathscr{T}$, for the agent concerned, but that the transformation of $\mathscr{R}$ (instead, or as well) may also be a valid response, noting that such a transformation may be non-trivial in a multi-agent environment.

To transform the set $\mathscr{T}$ in a useful way, we need to identify one or more valued concepts, in the conceptual space constrained by $\mathscr{R}$ (otherwise, we may introduce aberration – see below), and to use them to guide the transformation. However, there is a methodological problem here: there is no clear way to pick the concept(s) automatically, except at random or by use of an oracle. The 'oracle' might in fact be systematic search of $\mathscr{R}$ (assuming this is possible in finite time) or, again, the *deus ex machina* of user intervention.

There are some interesting issues to be considered here about the dynamics of this aspect of a creative system. There are obvious possibilities in analogy with the development of creative thinking through education. These, however, are outside the scope of the current chapter.

2.5.2.2 Aberration

Now, consider the following more interesting scenario, which also concerns the relationship between $\mathscr{R}$ and $\mathscr{T}$. A creative agent, **G**, is traversing its conceptual space. From any (partial) concept(s) in the conceptual space, **G**'s technique will enable it to create other concepts. Suppose now that the new concepts do not conform

to the constraints required for membership of the existing conceptual space,[11] and are therefore not selected by $[\![\mathscr{R}]\!](.)$. In this case, the set $\mathscr{B}$ given by

$$\mathscr{B} = \langle\langle \mathscr{T}, \mathscr{R}, \mathscr{E} \rangle\rangle^{\diamond}(\{\top\}) \setminus [\![\mathscr{R}]\!](\mathscr{U})$$

is non-empty. I term this *aberration*, since it is a deviation from the notional norm as expressed by $\mathscr{R}$. The choice of this rather negative terminology is deliberate, reflecting the conservatism with which changes to accepted styles are often met in the cultural world.

The evaluation of this set of concepts is actually slightly more complicated than the single-concept motivating case outlined above. The aberrant but valued subset, which I call $\mathscr{V}$ here, is computed thus:

$$\mathscr{V} = [\![\mathscr{E}]\!](\mathscr{B}).$$

Because we are working in the extensional limit case, with all the created concepts notionally elaborated, we have to consider the possibility that all aberrant concepts, some aberrant concepts or no aberrant concepts may be valued. I term these *perfect* ($\mathscr{V} = \mathscr{B}$), *productive* ($\mathscr{V} \subset \mathscr{B}$) and *pointless* ($\mathscr{V} = \emptyset$) aberration, respectively.

The notion of aberration is defined *within* the creative system, though an equivalent idea can evidently be applied from the outside. Thus, it affords a mechanism by which unexpected results can be detected and dealt with, as explained below. In this volume, Grace and Maher (2019) discuss a more nuanced notion of unexpectedness in creativity, very similar to that studied, in a broader theory of the information dynamics of cognition, by Pearce and Wiggins (2012).

Perfect aberration ($\mathscr{V} = \mathscr{B}$) yields new concepts, all of which are valued, and so should be added to $\mathscr{R}$. $\mathscr{T}$ has enlightened us as to new possibilities. We therefore attempt to revise $\mathscr{R}$, by whatever learning methods are available, in such a way that all the concepts in $\mathscr{B}$ (and $\mathscr{V}$) are included, so $\mathscr{V}$ is a positive training set, and the negative training set is either $\emptyset$ or $\mathscr{U} \setminus [\![\mathscr{R}]\!](\mathscr{U}) \setminus \mathscr{B}$ or some subset of the latter, depending on the effect desired. This, of course, is subject to the same caution as conceptual uninspiration above: if $\mathscr{R}$ is a representation of an agreed domain between multiple agents, then we need agreement on changing it; the same issue arises in the definition of (any concrete) $\mathscr{E}$. Again, however, these issues are beyond the scope of the current chapter.

Productive aberration ($\mathscr{V} \subset \mathscr{B}$) means that we need to transform both $\mathscr{R}$ and $\mathscr{T}$, because we wish valued concepts to become accepted, and unvalued ones not to be generated. $\mathscr{V}$ and $\mathscr{B} \setminus \mathscr{V}$ constitute positive and negative training sets for $\mathscr{R}$, since $\mathscr{R}$ needs to expand just enough to include only the valued concepts in $\mathscr{B}$. $\mathscr{T}$, on the other hand, needs to be transformed to restrict its coverage: $\mathscr{B} \setminus \mathscr{V}$ is a negative training set for $\mathscr{T}$, while, again, a positive training set might be $[\![\mathscr{R}]\!](\mathscr{U})$, or simply $\emptyset$.

[11] Note that there is no guarantee that they should do so – there is only an implicit assumption in Boden's work.

Pointless aberration ($\mathscr{V} = \emptyset$) suggests the need to transform $\mathscr{T}$ only, so as to prevent the unvalued aberrant concepts from being generated. There is a negative training set: $\mathscr{B}$. Again, the nature of the positive training set is an open question.

2.5.3 Discussion

These labels and definition allow us to characterise the behaviour of a given creative system and to identify broad classes of response. This, in turn, will allow comparison of behaviours both between and within the classes defined above, and thus allow better understanding of the field.

The emphasis in this work is on the further definition and understanding of the three sets $\mathscr{R}$, $\mathscr{T}$ and $\mathscr{E}$ and their relationships to each other, to the creative domain and to the activity they are intended to describe. In any case, what does become clear when one looks in detail at these proposals is that Boden's originals were (intended to be) rather broad-brush, and that when one focuses in, the relationships between the conceptual space, evaluation and the universe (albeit only implicit in Boden's work) become less, not more, simple.

Three clarifications do seem to emerge naturally from this discussion. First, to be interesting, $\mathscr{R}$ must define a set which is in some sense external to a given creative agent (and, I have supposed, agreed within an agent society – for whence else would it come?); second, $\mathscr{T}$ is the primary characterisation of the agent itself, and in this context, $\mathscr{R}$ is secondary (as in aberration, above); and, third, $\mathscr{E}$ needs to be independent of $\mathscr{R}$. This last needs a little elucidation, since, at first sight, it looks like a contradiction. The point is that, for transformational creativity to occur, there needs to be aberrant behaviour (unless we allow arbitrary spontaneous behaviour from our agents, which seems inappropriate). Otherwise, unless $[\![\mathscr{R}]\!](\mathscr{U})$ is infinite, the creative behaviour will stagnate, and the system will develop no further. While this is, of course, likely to be true of AI creative systems in the foreseeable future, it would be unfortunate if they were condemned to be so for all eternity. We can explain the apparent contradiction as follows: the set $[\![\mathscr{R}]\!](\mathscr{U})$ is specific to the domain, and effectively defines it. But the set constrained by $\mathscr{E}$ need only be the extension in $\mathscr{U}$ of *those properties of* $[\![\mathscr{R}]\!](\mathscr{U})$ *which are valued*. Thus, $[\![\mathscr{E}]\!](\mathscr{U})$ could be very large, but only a small part of it might be explored, because of the restrictions in $\mathscr{R}$ and $\mathscr{T}$.

The issue of multi-agent creative systems is becoming increasingly important in the current line of reasoning. The aim of Boden's and my frameworks is to describe the behaviour of creative systems, but no natural creative systems exist in isolation (and, indeed, one might argue that neither do artificial ones). Therefore, the generalisation of these ideas, which has been informally mentioned above on several occasions, to multi-agent systems seems crucial and urgent. Only in this context will the distinctions highlighted above become really clear, as the shared and individual content in the system will need to be made explicit. Some ground-breaking work in this direction has already been published (A. A. Kantosalo, Toivanen, Toivonen, et al.,

2015; A. Kantosalo, Toivanen, Xiao, & Toivonen, 2014; A. Kantosalo & Toivonen, 2016; Wiggins & Forth, 2017)

2.6 Illustration: One Millennium of Music

2.6.1 Introduction and Disclaimer

Now let us consider an example. An important caveat: I do not claim that this is in any sense *the single correct* characterisation of the domain in question. It is merely a simplistic illustration, but one which I find quite useful. The example is the development of Western art music throughout the second millennium, common era. Readers wishing to follow up the historical data here may refer to Abraham (1979). A useful dictionary of musical terms and concepts (and much more) is given by Scholes (1970).

2.6.2 Definitions

First, we need some definitions:

$\mathscr{U}$: All possible (partial and complete) pieces of music.
$\mathscr{L}$: A language for defining musical constraint and construction rules.
$[\![.]\!]$: An interpreter for selecting musical pieces from $\mathscr{U}$ according to rule sets specified in $\mathscr{L}$.
$\langle\langle .,.,. \rangle\rangle$: A search engine for traversing $\mathscr{U}$ and its subsets according to rule sets specified in $\mathscr{L}$.
$\mathscr{R}_S$: The rules for composition of music in style S.
$\mathscr{T}_C$: The rules defining the technique of composer C.
$\mathscr{E}_p$: The rules defining the preference of person p.

We can add, for convenience, the conceptual spaces $\mathscr{C}_S$, each of which contains all the possible (partial) pieces of music in style S, selected from $\mathscr{U}$ by $[\![\mathscr{R}_S]\!]$. In fact, it will not be necessary to use all of these definitions in this broad-brush example, but it is nevertheless useful to understand how the whole framework is constructed.

2.6.3 The Dark Ages and the Proto-Renaissance

Little is known about music in the Middle East and West in the period after the decline of ancient Greek society and before around 800–850. Thereafter, the majority of what is known is church music, nearly all folk or popular music being passed on by oral tradition and now, therefore, lost or changed.

We begin to see more formal, notated music in the 10th century, again mostly from the church. Those limited manuscripts which are available contain almost exclusively monophonic or drone-based modal melodies, or occasionally melodies sung in mostly parallel intervals, known as *organum*.[12] This is the starting point of our creative simulation. We need a set of rules, $\mathscr{R}_{\text{Modal}}$, which define that subset of all possible pieces of music which are in the *modal* style[13]. This gives us, in turn, $\mathscr{C}_{\text{Modal}}$, which is (one way of expressing) the conceptual space of modal music.

The exact nature of the music – monophonic, drone-based, organum – and the different styles of different composers are appropriately modelled by different sets $\mathscr{T}_C$, traversing the same conceptual space in different ways. But those rules are so constructed as never to reach the music of later periods, involving, for example, true polyphony. So, much of $\mathscr{C}_{\text{Modal}}$ is uncharted at the beginning of this period, with most composers covering and covering again a small subset.

However, by the time of the so-called Proto-Renaissance (*c.*, 1125–), the beginnings of three- and four-part harmony are emerging. These developments all take place within the well-established framework of C_{Modal}. Throughout this period, successive composers explore successive parts of the conceptual space with their individual $\mathscr{T}_C$s – the overall effect is one of a single creator exploring C_{Modal} with one grand, inclusive $\mathscr{T}$, though, of course, the actual process may be much more complex, with many steps and transformations, as suggested earlier.

The music is, however, still very simple and restrained, with, for example, major thirds still being regarded as dissonant intervals requiring resolution: scores invariably end in unison or open fifths.

2.6.4 Ars Nova

During the 14th century, we see the establishment of true polyphony in music, notably with the French composer Guillaume de Machaut. From the point of view of our model, there is little to add here, other than 'more of the same'. The point is that we are amidst a sustained period of exploration of $\mathscr{C}_{\text{Modal}}$.

2.6.5 The Renaissance period

By the 15th century, more of the modal space has been explored, in different ways, leading to more chordally accompanied musics, or homophony, and in another direction, more development of polyphony.

[12] Abraham (1979) refers to the homophonic *organa* of Hucbald, from the late 800s as polyphony, though this definition is debatable.

[13] Modal music, in fact, goes back at least as far as the Ancient Greeks, to the writings of Aristoxenus and Pythagoras, so as it has endured over 1000 years to this point in time, we can argue that it is a good basis for discussion.

Another strong trend during this time is towards a richer harmony – for example, cadences now often contain major or minor thirds, which were previously considered dissonant and therefore non-final.

All of this may be said to be achieved by exploratory creativity – the rules, $\mathscr{R}_{\text{Modal}}$, governing the fundamental nature of harmony are not in fact changing, but the subsets of $\mathscr{C}_{\text{Modal}}$ explored by different composers are becoming larger.

A related question is raised, however, by the change in the perception of dissonance mentioned above. While it is clearly the case that the exploration of $\mathscr{C}_{\text{Modal}}$ *admits* these new sounds, it is the evaluation function, $\mathscr{E}$, which *keeps* them. There is an interesting question to be considered of how 'learning' interaction between the $\mathscr{E}_p$s and the $\mathscr{T}_C$s works, since there clearly has to be such an interaction for this kind of development to proceed. This question is studied by A. Kantosalo et al. (2014), A. Kantosalo and Toivonen (2016) and Wiggins and Forth (2017).

2.6.6 The Baroque Period

With the advent of the well-tempered scale[14] in the time of J. S. Bach, a curious change takes place. It might be described as a transformation in the prevailing $\mathscr{T}_C$s, but I would argue that it is in fact a transformation of $\mathscr{R}_{\text{Modal}}$ into something else. At first sight, the effect might be seen as a strong tendency to explore a *more* limited part of $\mathscr{C}_{\text{Modal}}$ than before: that part corresponding with diatonic or tonal music, $\mathscr{C}_{\text{Tonal}}$. However, in fact, the change in tuning system has made it possible to use *more* scales which are diatonic. Previously, in just-tuning, the number of diatonic scales which could be played in tune was quite limited. The arrival of well-temperament meant that it was now possible to play in all keys without tuning problems; indeed, the effect of well-temperament is to give different keys distinctly different sounds, to a careful listener. This meant, in turn, that it was possible to achieve musical effects in diatonic ways which were previously only available via different modes. Because of the categorical perception of pitch, which motivates the use of well-temperament in the first instance, we may say that the diatonic members of $\mathscr{C}_{\text{Modal}}$ are in $\mathscr{C}_{\text{Tonal}}$ – but $\mathscr{C}_{\text{Tonal}}$ contains many pieces which cannot be in $\mathscr{C}_{\text{Modal}}$, because inappropriate tuning would render them unmusical.

Here, then, we are seeing our first significant transformational creativity: the explicit, deliberate change of tuning system and the new space of possibilities it enables have made a few key individuals change the $\mathscr{R}$s of their personal conceptual spaces – the change then propagates quickly through musical society and becomes collective. It seems likely that this change was achieved by understanding what was potentially possible given the right tuning system, and then finding the tuning system which would achieve it; in any case, the choice of the new system was explicitly chosen for its improved range, as in J. S. Bach's *The Well-Tempered Keyboard*.

[14] The common idea that well-temperament is the same as equal temperament, in which the modern piano is tuned, is incorrect. Well-tempered scales are a compromise, and themselves have interesting consequences for the sound of the music played.

2.6.7 The Classical Period: Comparing Creativity

The classical period sees more restriction in the diatonic conceptual space, refining the notion and use of dissonance in music still further. A particularly noteworthy pair of musicians at this point are Joseph Haydn and Wolfgang Amadeus Mozart. It is widely assumed that Mozart was 'the more creative' of the two, but, objectively, this is open to question. While history seems to suggest that Mozart produced 'better' music, it was nevertheless Haydn who really defined the classical style, which Mozart then improved. So $\mathscr{T}_{\text{Haydn}} \cap \mathscr{T}_{\text{Mozart}}$ is large compared with $\mathscr{T}_{\text{Haydn}} \cap \mathscr{T}_{C}$ where C is an earlier composer. It is easy to see that Haydn is in at least one sense more creative when the issues are thus expressed.

2.6.8 The Romantic Period

Notwithstanding the restriction from modal to diatonic music, the tendency to explore more harmonically dense structures continues, and chromaticism begins to emerge. This music is still tonal, but it is stretching the boundaries of what can be called tonality, often to the consternation of successive generations of audiences.

Nonetheless, the music is still essentially tonal, and so inhabits essentially the same conceptual space defined two thousand years earlier in ancient Athens. What has changed is the amount of the conceptual space, $\mathscr{C}_{\text{Tonal}}$, covered by the search of a collective $\mathscr{T}$ representing the sum of musical exploration and accepted by a collective $\mathscr{E}$ becoming progressively more tolerant of dissonance.

2.6.9 Modernist Music

The twentieth century saw the arrival of a new intellectualism in music, where experiment in method became as valued by some observers as much as or more than the creative output – so here we have an agreed change in $\mathscr{E}$, moving from a judgement on creative output to the same combined with a judgment on the nature of the creative process itself. This led to an explosion of styles, some retrospective, such as the modality of Vaughan Williams, and some quasi-retrospective, such as the neo-classicism of early Stravinsky, and some new, such as the experimentalism of Ives, Varèse and Cage. Here is another point, then, at which a change in $\mathscr{E}$ opens up a whole new area of conceptual space, and possibly of universe, for consideration. Indeed, many of the works in question would not even be considered as music under the definitions of earlier centuries.

2.6.10 Twelve-Note Music

Arguably the most radical change, however, arose with Arnold Schoenberg's Opus 11 in 1920, the first piece deliberately not centred on a key note.[15] The tone-centred assumption of modality and tonality finally is shattered – and, importantly, it is shattered consciously and deliberately, along with associated notions, such as dissonance.[16] This, then, in both Boden's terms and mine, is another transformation. The followers of Schoenberg created music which inhabits a different space of possibilities from $\mathscr{C}_{\text{Tonal}}$ – we might call it $\mathscr{C}_{\text{Twelve-note}}$. By definition, *complete* members of $\mathscr{C}_{\text{Twelve-note}}$ cannot be members of $\mathscr{C}_{\text{Tonal}}$.

The difference between $\mathscr{C}_{\text{Twelve-note}}$ and $\mathscr{C}_{\text{Tonal}}$ is sufficiently great that no $\mathscr{T}$ designed for the latter will work for the former – so the Second Vienna School composers were forced to develop their own explicit methodology, based around Schoenberg's 'twelve-note method'. Society is still waiting for an $\mathscr{E}$ agreed enough to allow these composers to enter mainstream popularity.

2.6.11 Summary

In this section, I have illustrated how the development of music from around the 10th century to the time of writing may be outlined using my proposed formalism, and have highlighted one or two places where doing so elucidates what was actually happening during that development. Clearly, however, there is much more work to be done in this area.

2.7 Summary and Conclusion

This chapter has presented a small step on the road to a more precise understanding of creative systems, both artificial and natural. I have presented a framework for characterising creative systems and shown that Boden's transformational creativity is actually exploratory creativity at the meta-level. I have given six categorisations of creative behaviour, which can be identified directly from the behaviour of creative systems as described using the formalism, and suggested how the needs of each category of system can be met, from within or from outside the system itself. This raises many questions, not least the issue of interaction between multiple creative agents. These questions will be addressed in future work.

---

[15] Schoenberg did not use the term *atonal* – rather, he preferred *twelve-note*, presumably because he understood the Western tendency to perceive and understand music tonally even when it is not intended to be so.

[16] 'Dissonance is an outmoded concept,' (Schoenberg, 1974).

Acknowledgements I am very grateful to my colleagues in the Computational Creativity Lab at Queen Mary University of London for ongoing discussion about this and other work, and to the computational creativity community (especially Simon Colton and Penousal Machado), which has given much useful feedback in its various workshops and symposia. Graeme Ritchie supplied some particularly helpful comments on earlier related work, during his sabbatical at City University, London, in 2002, and in subsequent papers (Ritchie, 2012). Several anonymous reviewers gave some very helpful feedback on this and earlier versions.

References

Boden, M. A. (1977). *Artificial intelligence and natural man*. Harvester Press.

Boden, M. A. (1995). Modelling creativity: Reply to reviewers. *Journal of Artificial Intelligence*, *79*, 161–182.

Boden, M. A. (1998). Creativity and artificial intelligence. *Artificial Intelligence Journal*, *103*, 347–356.

Boden, M. A. (1999). Preface to special issue on creativity in the arts and sciences. *AISB Quarterly*, *102*.

Boden, M. A. (2004). *The creative mind: Myths and mechanisms* (2nd). London: Routledge.

Buchanan, B. G. (2001). Creativity at the metalevel. *AI Magazine*, *22*(3), 13–28. AAAI-2000 presidential address. Retrieved from http://www.aaai.org/Resources/President/president.html

Bundy, A. (1994). What is the difference between real creativity and mere novelty? *Behavioural and Brain Sciences*, *17*(3), 533–534.

Colton, S., & Wiggins, G. A. (2012). Computational creativity: The final frontier? In L. de Raedt, C. Bessiere, D. Dubois, & P. Doherty (Eds.), *Proceedings of ECAI frontiers*. doi:10.3233/978-1-61499-098-7-21

Correia, J., Machado, P., Romero, J., Martins, P., & Cardoso, F. A. (2019). Breaking the mould: An evolutionary quest for innovation through style change. In T. Veale & F. A. Cardoso (Eds.), *Computational creativity: The philosophy and engineering of autonomously creative systems* (pp. 353–399). Springer.

Gervás, P. (2002a). Exploring quantitative evaluations of the creativity of automatic poets. In C. Bento, A. Cardoso, & G. A. Wiggins (Eds.), *Proceedings of the ECAI'02 workshop on creative systems: Approaches to creativity in AI and cognitive science* (pp. 39–46). Lyon, France.

Gervás, P. (2002b). Linguistic creativity at different levels of decision in sentence production. In A. Cardoso & G. A. Wiggins (Eds.), *Proceedings of the AISB'02 symposium on AI and creativity in arts and science* (pp. 79–88). www.aisb.org.uk: AISB.

Grace, K., & Maher, M. L. (2019). Expectation-based models of novelty for evaluating computational creativity. In T. Veale & F. A. Cardoso (Eds.), *Computational creativity: The philosophy and engineering of autonomously creative systems* (pp. 193–207). Springer.

Haase, K. (1995). Too many ideas, just one word: A review of Margaret Boden's *The Creative Mind: Myths and Mechanisms*. *Artificial Intelligence Journal*, *79*, 69–82.

Kantosalo, A. A., Toivanen, J. M., Toivonen, H. T. T., et al. (2015). Interaction evaluation for human-computer co-creativity. In *Proceedings of the sixth international conference on computational creativity*.

Kantosalo, A., Toivanen, J. M., Xiao, P., & Toivonen, H. (2014). From isolation to involvement: Adapting machine creativity software to support human-computer co-creation. In *Proceedings of the fifth international conference on computational creativity* (pp. 1–8).

Kantosalo, A., & Toivonen, H. (2016). Modes for creative human-computer collaboration: Alternating and task-divided co-creativity. In *Proceedings of the seventh international conference on computational creativity*.

Koestler, A. (1976). *The act of creation*. London: Hutchinson.

Lavrač, N., Juršič, M., Sluban, B., Perovšek, M., Urbančič, T., & Cestnik, B. (2019). Bisociative knowledge discovery for cross-domain literature mining. In T. Veale & F. A. Cardoso (Eds.), *Computational creativity: The philosophy and engineering of autonomously creative systems* (pp. 119–138). Springer.

Lustig, R. (1995). Margaret Boden, *The Creative Mind: Myths and Mechanisms*. *Artificial Intelligence Journal*, *79*, 83–96.

Macedo, L., & Cardoso, A. (2001). Creativity and surprise. In *Proceedings of the AISB'01 symposium on creativity in the arts and sciences* (pp. 84–92). Brighton, UK: SSAISB.

Abraham, G. (1979). *The concise Oxford history of music*. Oxford, UK: Oxford University Press.

Pearce, M. T., & Wiggins, G. A. (2001). Towards a framework for the evaluation of machine compositions. In *Proceedings of the AISB'01 symposium on artificial intelligence and creativity in the arts and sciences* (pp. 22–32). Brighton, UK: SSAISB. Retrieved from http://www.soi.city.ac.uk/~ek735/papers/aisb01.pdf

Pearce, M. T., & Wiggins, G. A. (2012). Auditory expectation: The information dynamics of music perception and cognition. *Topics in Cognitive Science*, *4*(4), 625–652.

Pérez y Pérez, R. (2019). Representing social common-sense knowledge in MEXICA. In T. Veale & F. A. Cardoso (Eds.), *Computational creativity: The philosophy and engineering of autonomously creative systems* (pp. 253–272). Springer.

Perkins, D. (1995). An unfair review of Margaret Boden's *The Creative Mind* from the perspective of creative systems. *Artificial Intelligence Journal*, *79*, 97–109.

Ram, A., Wills, L., Domeshek, E., Nersessian, N., & Kolodner, J. (1995). Understanding the creative mind: A review of Margaret Boden's *Creative Mind*. *Artificial Intelligence Journal*, *79*, 111–128.

Ritchie, G. (2007). Some empirical criteria for attributing creativity to a computer program. *Minds and Machines*, *17*(1), 67–99.

Ritchie, G. (2012). A closer look at creativity as search. In *Proceedings of the 3rd international conference on computational creativity*, Dublin, Ireland.

Ritchie, G. (2019). The evaluation of creative systems. In T. Veale & F. A. Cardoso (Eds.), *Computational creativity: The philosophy and engineering of autonomously creative systems* (pp. 157–192). Springer.

Russell, S., & Norvig, P. (1995). *Artificial Intelligence: A Modern Approach*. Eaglewood Cliffs, NJ: Prentice Hall.

Schank, R., & Foster, D. (1995). The engineering of creativity: A review of Boden's *The Creative Mind*. *Artificial Intelligence Journal*, *79*, 129–143.

Schoenberg, A. (1974). *Letters*. Edited by Erwin Stein. Translated from the original German by Eithne Wilkins and Ernst Kaiser. London: Faber.

Scholes, P. A. (1970). *The Oxford companion to music* (10th). Oxford, UK: Oxford University Press.

Turner, S. R. (1995). Margaret Boden, *The Creative Mind*. *Artificial Intelligence Journal*, *79*, 145–159.

Wiggins, G. A. (2006a). A preliminary framework for description, analysis and comparison of creative systems. *Journal of Knowledge Based Systems*, *19*(7), 449–458. Retrieved from http://dx.doi.org/10.1016/j.knosys.2006.04.009

Wiggins, G. A. (2006b). Searching for computational creativity. *New Generation Computing*, *24*(3), 209–222.

Wiggins, G. A., & Forth, J. C. (2017). Computational creativity and live algorithms. In R. T. Dean & A. McLean (Eds.), *The Oxford handbook of algorithmic music*. Oxford University Press.

Wiggins, G. A., Tyack, P., Scharff, C., & Rohrmeier, M. (2015). The evolutionary roots of creativity: Mechanisms and motivations. *Philosophical Transactions of the Royal Society of London B: Biological Sciences*, *370*(1664). doi:10.1098/rstb.2014.0099

Chapter 3
Autonomous Intentionality in Computationally Creative Systems

Dan Ventura

Abstract We consider the problem of building an artificial system that exhibits creativity. We begin by noting that the attribution of creativity is influenced by perceptions about the system and that perhaps the most difficult hurdles to overcome in this respect are those of system *intentionality* and system *autonomy*. We discuss ways in which these difficulties might be addressed and cite examples from the DARCI system that already exhibit these characteristics to some extent. We also discuss the difficult problem of *evaluation* and present several useful methods, as well as examples of their application to results produced by DARCI.

3.1 Introduction

It has been argued that the attribution of creativity depends not only on the demonstration of characteristics like novelty, value and surprise but also to a large extent on the impression that the creative act was "done the right way" (Colton, 2008). We performed a Wizard-of-Oz experiment recently that validates this argument and helps us better understand what "done the right way" might mean in the case of computational systems.

The experiment consisted of participants interacting with what they were told was an analogy-making program with (up to) three levels of sophistication (in reality, the source of the analogies was a person at a remote terminal). Participants were asked at random either to participate in one of the three levels, or to move through all three consecutively. After observing the analogies that were generated by the "program", they were asked to evaluate the creativity of the system, as well as to determine where they felt the attribution of creativity belonged on a five-point Likert scale from programmer to program. We observed that as the degree of interactive sophistication increased, participants' willingness to attribute creativity toward the program (and

Dan Ventura
Computer Science Department, Brigham Young University. e-mail: ventura@cs.byu.edu

T. Veale, F. A. Cardoso (eds.), *Computational Creativity*, Computational Synthesis and Creative Systems, https://doi.org/10.1007/978-3-319-43610-4_3

away from the programmer) also increased. We also observed that those who tried all three successively more sophisticated interaction levels attributed dramatically more creativity to the system (more so than those participating in individual tests), perhaps because they were (surprisingly) forced to revise their own assessments multiple times. We conducted three surveys among different audiences (Reddit users, software engineers and computer science academics), asking about computers and creativity. Each participant was asked to rate whether computers were currently capable of creativity, and whether they will someday be capable of creativity, on a Likert scale from 0 to 10. They were then asked to define what they thought were essential requirements for or characteristics of creative attribution. Finally, they were asked to describe what behavior or characteristics a system should have in order to convince them that it was creative. The academics as a group were the most skeptical, while the programmers as a group were much more accepting of the idea of a computational system being creative. Regarding essential characteristics, while participants mentioned variations on most of the creative attributes typically discussed in the field (appreciation, skill, novelty, typicality, learning, individual style, curiosity, accountability), particularly among the most skeptical participants (those who rated it unlikely that computers are or ever will be creative), autonomy and intentionality were the top priorities for creativity (Mumford & Ventura, 2015).

To summarize the results, in order to be persuaded that a computational system is creative, respondents wanted evidence that the system can "think for itself." In other words, they require a convincing *evaluation* of the *intentionality* and the *autonomy* of the system. With this in mind, we will attempt to show that it *is* possible to create artificial systems that are both intentional and autonomous and also that these characteristics can be demonstrated using sound methods of evaluation. For concrete examples supporting our arguments, we will make use of results from the DARCI system (Heath, Norton, & Ventura, 2013; Norton, Heath, & Ventura, 2010, 2011b, 2013) which operates in the visual art domain, but note that we have demonstrated many of these points in computational systems operating in other domains as well, including music (Johnson & Ventura, 2016; Murray & Ventura, 2012), games (Lebaron, Mitchell, & Ventura, 2015), cooking (Morris, Burton, Bodily, & Ventura, 2012), and language (Smith, Hintze, & Ventura, 2014).

We proceed first by briefly introducing the DARCI system. We then argue that intentionality can be demonstrated by the communication of concepts via a common perceptual grounding. Next, we argue that autonomy can be exhibited by demonstrating imagination, making use of inspirational sources and creating meta-level artefacts. Finally, we discuss some ways these claims might be evaluated for computational systems and offer some brief concluding remarks.

3.2 DARCI

DARCI is a system for generating original images that convey intention and is inspired by other artistic image generating systems such as AARON (McCorduck,

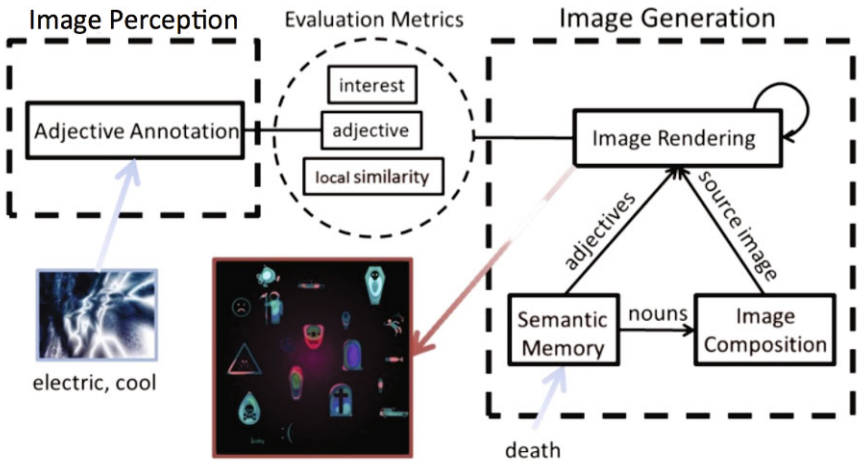


Fig. 3.1 A functional description of DARCI. *Image perception* uses neural models to ground semantic knowledge (acquired from, e.g., image labels, semantic networks and vector space modeling) using a set of 51 general computational vision features. *Image generation* uses evolutionary computation to produce a visual representation that communicates concepts (through both content and rendering style) that are semantically cohesive and related to a target idea. The two components interact through a set of evaluation metrics to ensure that the generated image is perceptually grounded.

1991) and The Painting Fool (Colton, 2011). Central to the design philosophy of DARCI is the notion that the communication of meaning in art is a necessary part of eliciting an aesthetic experience in the viewer (Csikszentmihalyi & Robinson, 1990), and it is unique in that it creates images that intentionally express a given concept using *visual metaphor*.

DARCI is composed of two major components, an *image perception* component, and an *image generation* component. The image perception component learns how to associate images with semantic knowledge. The image generation component composes an original source image to visually convey a concept; see Fig. 3.1.

3.2.1 Image Perception

DARCI employs three main approaches to perception: *visuo-linguistic association*, a *semantic network* and a *vector space model*. These allow the system to ground concepts in terms of both images and other concepts.

3.2.1.1 Visuo-linguistic Association

In order for DARCI to make associations between images and concepts, the system learns a mapping from 51 low-level computer vision features to words, using a database of labeled images. These features can be grossly categorized as color features (standard color space statistics, emotional color space statistics, rule of thirds color statistics, colorfulness, emotional histograms, color histogram); global texture and shape features (co-occurrence, edge frequency, primitive length, image moments, edge direction histogram); and summary-of-local-interest-point features (interest point statistics, principal components, SURF descriptor statistics, bag-of-visual-words data).

To collect training data we have created a public website for DARCI[1], where users are presented with random images and asked either to provide adjectives that describe the image or to indicate which adjectives that DARCI currently associates with the image are not correct. The result is a dataset representing a difficult, high-dimensional multilabel classification problem that we refer to as the DARCI dataset.

DARCI uses a collection of artificial neural networks called *appreciation networks*, one for each unique adjective, that output a single real value, between 0 and 1, indicating the degree to which a given image can be described by the network's associated adjective. An appreciation network is created for each adjective that has a sufficient amount of training data, and as data is incrementally accumulated, new networks are dynamically added to the collection to accommodate any new adjectives.

3.2.1.2 Semantic Network

The system also contains a semantic network forming a graph of associations between words. These word associations are acquired in one of two ways: from people and by automatic inference from a corpus; the human word associations capture general knowledge and the corpus associations augment this by "filling in the gaps" and providing less common associations.

For the human word associations, we use two pre-existing databases of *free association norms* (FANs): the Edinburgh Associative Thesaurus (Kiss, Armstrong, Milroy, & Piper, 1973) and the University of Florida's Word Association Norms (Nelson, McEvoy, & Schreiber, 1998). These word associations were acquired by asking hundreds of human volunteers to provide the first word that comes to mind when given a cue word. This technique is able to capture many different types of word associations including word co-ordination, collocation, super-ordination, synonymy, and antonymy. The association strength between two words is simply a count of the number of volunteers who said the second word given the first word. For the corpus-based associations, we build a (term $\times$ term) co-occurrence matrix from the (English) text of Wikipedia, as it is large and easily accessible, and covers a wide

[1] http://darci.cs.byu.edu

range of human knowledge (Denoyer & Gallinari, 2006). Once the co-occurrence matrix is built, we use the co-occurrence values themselves as association strengths between words.

The final semantic network is a composition of the human- and corpus-based associations, which essentially merges the two separate graphs into a single network. To combine the graphs, we add the top *n* associations for each word from the corpus data to the human data but weight the corpus-based association strengths lower than the human-based associations. This is beneficial for two reasons. First, if there are any associations that overlap, adding them again will strengthen the association in the combined network. Second, corpus-based associations not present in the human data will be added to the combined network and provide a greater variety of word associations.

3.2.1.3 Vector Space Model

DARCI also uses a vector space model called the *skip-gram model* (Mikolov, Chen, Corrado, & Dean, 2013) – a neural architecture that analyses a large corpus and learns to predict the surrounding words given a current word. During training, the skip-gram model consequently learns vector representations for each word, which encode semantic information. Words similar in meaning will have vectors that are close to each other in vector space.

These semantic vectors allow DARCI to find concepts related to a given word and to assess the similarity in meaning between words. In order for DARCI to leverage this information for image creation, it must learn to associate image qualities with the semantic vectors, analogously to the visuo-linguistic association discussed above. Currently, we limit the associated words to vectors representing adjectives, and use a neural network model to predict the adjective vector for a given image. Two separate neural networks are trained, one with positively labeled images, and one with negatively labeled images. The positive network tries to predict what adjective an image *is*, while the negative network tries to predict what adjective an image *is not*. Together, these networks learn to predict the appropriate adjective *vector* given an image.

Learning to predict an adjective's vector is a harder task than learning to predict the adjective directly, so the vector space model may not predict as accurately as using separate models for each adjective; however, the main advantage of learning the vectors is that we can do zero-shot prediction – DARCI is now not limited to the adjectives for which it was explicitly trained, rather it can predict vectors for words it has never learned, including nonadjectives.

3.2.1.4 Image Generation

DARCI's generative ability is effected by two main subprocesses: *composition*, dealing mainly with content, and *rendering*, dealing mainly with style.

3.2.1.5 Image Composition

The semantic memory and vector space models can be considered to represent the meaning of a word as a (weighted) collection of other words. DARCI effectively makes use of this collection as a decomposition of a (high-level) concept into simpler concepts that together represent the whole; the idea being that in many cases, if a (sub)concept is simple enough, it can be represented visually with a single icon. To represent these "simple enough" concepts, DARCI makes use of some collection of icons provided by, for example *The Noun Project* (Thomas, Boatman, Polyakov, Mumenthaler, & Wolff, 2013). Given such a collection of iconic concepts, DARCI composes their visual representations (icons) into a single image.

For example, consider the abstract concept 'war'. The semantic memory would discover related nouns like 'soldier', 'army', 'conflict', and 'battle' and related adjectives like 'bloody', 'violent', and 'lonely'. The nouns are sent to the image composer, which produces a *source* image by composing simple iconic images of these related words. The adjectives and this source image are then sent to the image renderer, which renders the source in a style that communicates one or more of the related adjectives. DARCI can also forgo composing an original source image and instead re-render an existing source image in an artistic way that expresses a particular concept.

3.2.1.6 Image Rendering

DARCI uses an evolutionary mechanism to render images so that their perceptual characteristics communicate a particular concept or set of concepts. The genotypes that comprise each gene pool are lists of filters (and their accompanying parameters) for processing a source image. The processed image is the phenotype. Evolutionary pressure towards perceptual grounding is applied by the visual semantic model, with the fitness function being of this form:

$$\text{Fitness}(f^P) = \lambda_A A(f^P) + \lambda_S S(f^P) \tag{3.1}$$

where P is the phenotype image and f^P is the vector of image features for a given phenotype, and $A : F^P \to [0,1]$ and $S : F^P \to [0,1]$ are two functions for modeling appreciation and similarity, respectively. These functions compute a real-valued score for a given phenotype (here, F^P represents the set of all phenotype feature vectors and $\lambda_A + \lambda_S = 1$).

The appreciation function A is some function of the perceptual model (appropriate appreciation network(s), semantics network, vector space model). The similarity function S borrows from research on *bag-of-visual-word* models (Csurka, Dance, Fan, Willamowski, & Bray, 2004; Sivic, Russell, Efros, Zisserman, & Freeman, 2005) to effectively measure the number of interest points shared by the two images and can be translated into a measure of image similarity.

Because there are potentially many ideal filter configurations for modeling a given concept, speciation is used within each gene pool, allowing the evolutionary mechanism to converge to multiple solutions. Limited migration between sub-populations is allowed, with the exception that the most fit genotype per sub-population is not allowed to migrate, and sub-population size balancing is enforced.

3.3 Intention

A system that exhibits intention is deliberative or purposive; that is, the output of the system is the result of the system having a goal or objective – the system's product is correlated with its process. This is perhaps best demonstrated through some form of communication. The form of the intention and the form of the communication will depend on the domain in question; for example, usually, in the domain of visual art, an artefact (typically an image of some kind) is intended to communicate some concept or idea. The communication of a concept requires an understanding of that concept, and this, in turn, requires some form of grounding, which is learned through perceptual experience. Indeed, this is consistent with Csíkzentmihályi's notion that "... the aesthetic experience occurs when information coming from the artwork interacts with information already stored in the viewer's mind ..." (Csikszentmihalyi & Robinson, 1990).

Thus, a realization of intention occurs when the system produces an artefact with the goal of communicating a particular concept and the consumer understands the concept as a result of being exposed to that artefact. This is possible if the creator and consumer share a common perceptual grounding – both understand the world in the same way.

3.3.1 Perception-Based Understanding

DARCI's understanding of images is derived from two sources: a mapping from low-level image features to descriptive adjectives and semantic associations between linguistic concepts. Fig. 3.2 shows the extent to which DARCI "understands" (using its appreciation networks) a collection of 110 adjectives.

By incorporating a sophisticated semantic model grounded in perception, DARCI is able to internally represent the meaning of concepts, which facilitates the expression of these concepts through images in innovative ways. DARCI's semantic model incorporates neural networks that have been trained to assess the aesthetic quality, the artistic style, and the semantic content of images. This allows DARCI to evaluate and understand its own artefacts, which in turn allows it to produce semantically relevant and aesthetically pleasing images. It also allows DARCI to evaluate other existing images in order to find inspiration and generate its own semantically related images in response.

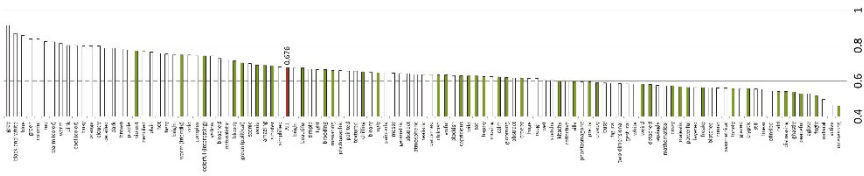


Fig. 3.2 The true rate of 110 adjectives from the DARCI dataset, using 51 forward-selected features. Affective adjectives are highlighted in green, and the average true rate of all 110 adjectives is indicated in red. True rate (TR) is the average of the true positive rate (TPR) and true negative rate (TNR) and is proportional to informedness (I), which is commonly used in psychology and has been shown to be less biased than other common measures including recall, precision, F-measure, and accuracy (Powers, 2011, 1). Since informedness is defined as $I = TPR + TNR - 1$, it follows that true rate is related to informedness as $I = 2 \cdot TR - 1$.

3.3.2 Communicating Intention

DARCI communicates ideas using visual metaphor, generating images in two stages: the creation of a source image composed of a collage of conceptual icons and the rendering of this source image using some set of parameterized image filters. The collage generation is driven by the semantic and vector space models, while the filtered rendering is achieved using an evolutionary mechanism for which the visual semantic model acts as the objective function. That is, the image is rendered in a style that is semantically consistent with the originating concept (based again in the vector space model's interpretation), i.e., during rendering, the visual semantic model acts as the fitness function to guide the rendering process. Fig. 3.3 shows several examples of photographs rendered by DARCI to communicate a particular concept, and Fig. 3.4 shows several examples of original images created by DARCI (to communicate a particular concept).

Another class of approaches to the question of intentionality involves system interaction (with another system or with a human) and the system's ability to "do the right thing" under a wide variety of conditions. If the variety and sophistication of the interactions are high enough, and if the system meets reasonable performance expectations, there is a greater likelihood that it may be considered intentional.

In addition, intention can also be communicated more directly if the system has an ability to explain its process and/or product – can the system justify in some way why it made the decisions it made and why the result is what it is? This is one example of the broad notion of *framing*, which conflates, to some extent, the notions of intentionality and autonomy. For additional discussion on this point and an example of DARCI using framing to directly communicate intention, see Section 3.4.2.

Fig. 3.3 Images that DARCI has rendered (bottom row) after being provided with a source image (top row) and a concept. From left to right, the concepts are 'fiery,' 'Alaska,' and 'hunchback.' Although the source image was given, DARCI discovered its own way to render the image to convey the given concept.

Fig. 3.4 Images that DARCI has created from scratch after being given only a concept. From left to right, the concepts are 'bizarre,' 'war,' 'art,' 'murder,' and 'hunger'.

3.4 Autonomy

Like creativity itself, autonomy is a slippery subject, but what is wanted is a system that can be seen as acting independently of its builder(s). Because this is an ill-defined idea, we will settle here for some less difficult (though certainly nontrivial) proxies: *imagination*, *inspiration* and *meta-level artefacts*. It is not difficult to argue that a system that demonstrates these will exhibit abilities beyond any explicitly provided by its designers, and thus enjoy some level of autonomy.

3.4.1 Imagination

Using a semantic memory that incorporates the models described in Section 3.2.1.1, DARCI can create images that communicate conceptual adjectives (and even nonadjectives) of which it has never seen example images. This ability is a rudimentary form of imagination and is analogous to a person being able to imagine, say, what a

'majestic' image might look like when told that 'majestic' is similar to 'powerful' and 'beautiful,' even though the person may have never experienced the word 'majestic' before.

This form of imagination is not limited to images and could be applied to practically any domain. For example, a system could generate music based on the same word vectors (e.g., compose a 'happy' song), and could then interpolate new music to match previously unheard concepts according to the semantic memory. This memory could even act as a bridge between different domains – the system could listen to a 'sad' song, which would be mapped near the 'sad' vector, and the system could then "imagine" a sad-like image inspired by the song.

Using vector-space-based semantic memory models for these types of creative tasks allows the system flexibility and autonomy, and demonstrates a rather robust form of intelligence. Such a semantic model attempts a form of transfer learning (e.g., from written text to image understanding/generation), enabling DARCI to imagine conceptually, even for concepts it has not learned. Fig. 3.5 shows several forms of imagination available to DARCI given its semantic memory structure.

3.4.2 Inspiration

A simple and effective way to imbue a system with additional autonomy is to have some unpredictable, yet nonrandom input to the system that affects its decision making in some nontrivial way. For example, a system might "read the news" and be affected by the sentiment of the articles it encounters in the form of a mood parameter that affects the tone and style of portraits created by the system (Colton & Ventura, 2014). Or, a system might use one image as a source for ideation of content, style, tone, etc. in the creation of another image. To be convincing, the inspiration derived by the system should not be predictable but at the same time should be defensible, ideally by the system itself. Consider the example of Fig. 3.6. Here, DARCI uses an image of the Mona Lisa (*La Joconde*) as inspiration for the creation of an original piece it titles, "Overdress", and, it is capable of explaining, both in rudimentary language and using images, how its finished work is related to the source of its inspiration.

3.4.3 Meta-level Artefacts

A system that can operate to some extent at the meta-level will necessarily have greater autonomy than one that cannot. A system that creates an image to convey a particular concept can be autonomous in how it represents that concept, but a system that decides *which* concept to convey is *ipso facto* more autonomous yet. One way to effect such autonomy is some form of inspiration, as discussed in Section 3.4.2. Another approach is meta-level creativity.

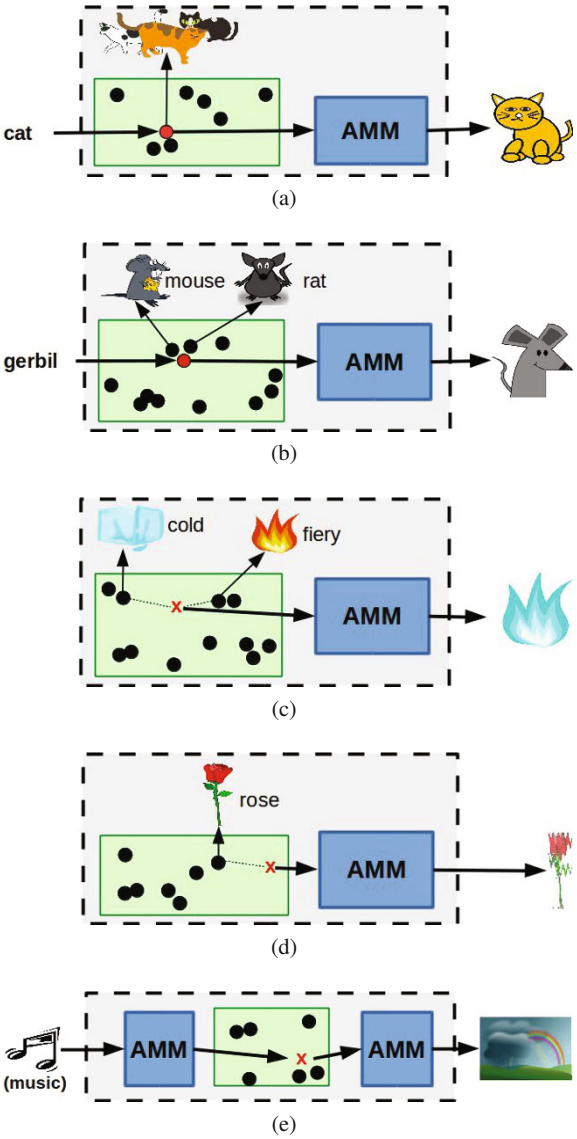
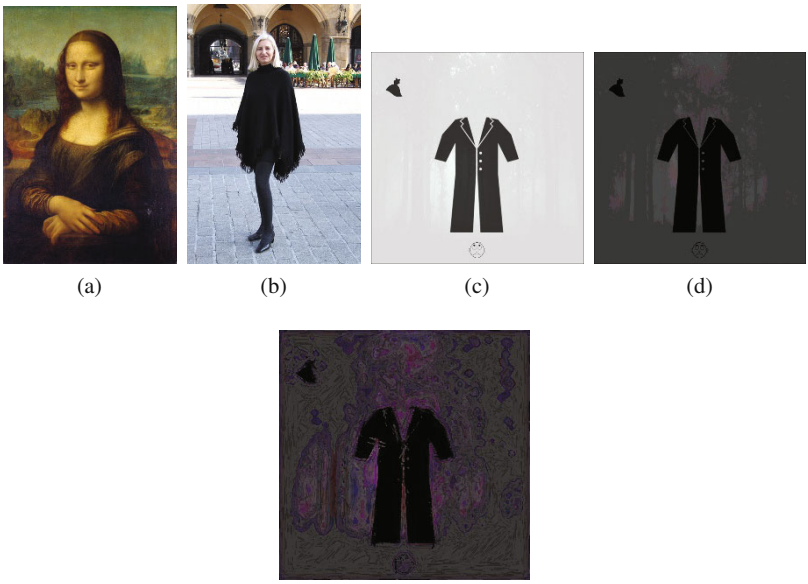


Fig. 3.5 Different ways a vector-space-based semantic memory framework can be used to imagine artefacts. The green rectangle with black dots represents concept vectors in conceptual space, which are learned by a vector space model. The Associative Memory Model (AMM) associates concept vectors with artefacts. The framework (a) allows the imagining of artefacts for concepts it has previously observed; (b) can facilitate the imagining of artefacts for concepts it has not previously observed but that are similar to other concepts that is has observed; (c) allows the imagining of artefacts that are combinations of two (or more) previously observed concepts; (d) can imagine changes to a previously observed concept; and (e) can facilitate imagination across different domains by observing an artefact in one domain and then imagining a related artefact in another domain.



(a) (b) (c) (d)

(e)

Fig. 3.6 An example showing DARCI's use and justification of inspiration in its image creation process (personifying the system): "I was looking for inspiration from this image (a), and it made me feel **gloomy** and **dreamy**. It also made me think of this image that I've previously seen (b), which is a picture of a **poncho**. So I started an initial image of my own by searching for a background image on the Internet based on **poncho**, **gloomy**, and **dreamy**. Then I took basic iconic images associated with those concepts and resized/placed them on the background according to how relevant they were. This was the result (c). I then modified it in a style related to **poncho**, **gloomy**, and **dreamy**, which resulted in this image (d). I did a final modification based on aesthetic quality and how closely the style related to the original image e). The end result perhaps looks more like a **cloak** or a **vestment**, and it feels particularly **gloomy**. I call it **Overdress**."

Perhaps the simplest way to make this possible is a system's use of a linearly weighted, multiobjective fitness function,

$$f(x) = \sum_{e \in E} \alpha_e FeatureScore(t_e, e(x))$$

parameterized by the set of targets $\{t_e\}$ and a set of weights $\{\alpha_e\}$. Here, x is the artefact being evaluated, E is a (fixed, for now) set of feature extractors, the t_e are the target feature values and the α_e weight the extractors. With this framework, a system can change its objective function by changing the values of the t_e and/or the α_e. This can be effected by considering a meta-level creative task for which fitness functions are the artefact class. With E fixed, these meta-artefacts can be represented as a pair of real-valued vectors, $\mathbf{t}$ and α. This, in turn, requires (at the least) some mechanism for the creation of candidate artefacts (a meta-level generative process for creating $\mathbf{t}$ and α) and some mechanism for evaluating them (a meta-level fitness function that

(a) (b) (c)

Fig. 3.7 Examples of DARCI creating different styles for image rendering – the system can create an appropriate style for communicating a particular concept. (a) combines a style for communicating *fear* with an image of a robot; (b) combines a style for *weird* with an image of a still life; and (c) combines a style for *frightening* with an image of a pig. Note that these styles are not predefined; DARCI creates them based on its perceptual understanding of the concepts.

says how "good" **t** and α are). Note that there is some interplay between the two sets of parameters but that they serve different functions: the targets control the *quality* of different features, while the weights control their *importance*.

As a concrete example, the set E could be a set of functions for computing various image features involving color, texture, shape, etc. with the set $\{t_e\}$ targeting particular values for those features (reddish, with a predominantly rough texture, with geometric lines, etc.). The set α_e weights each of these features (perhaps a reddish color is less important than a rough texture – see Murray and Ventura (2012) for details). Each different setting of the weights/targets then corresponds to some different visual style, with greater differences in weights/targets representing more marked differences in style. See Fig. 3.7 for examples of DARCI creating different styles to communicate different intentions.

A more ambitious program will let the set of feature extractors E be mutable as well, with the system able to remov or modify them or add feature extractors to E. Even more sophisticated meta-creativity can also be conceived (e.g., a fitness function for evaluating the interestingness of inspirational sources), though it becomes increasingly difficult to operationalize as the abstraction level increases.

3.5 Evaluation

If we are going to be serious about studying creativity in computational systems, and if we are going to claim autonomy and intentionality for such systems, it is critical that we have a way to evaluate our claims and to measure our progress. But, how do we measure such things? This is a notoriously difficult question that the field as a whole has yet to satisfactorily resolve in the general case. However, for systems for which we are claiming some level of semantic understanding and the ability to communicate intention, one obvious suggestion is to evaluate those particular claims, both absolutely and relationally.

Measuring a system's understanding and communication ability in an absolute sense means asking whether observers of the system can correctly interpret its intention given the cues provided by the system (e.g., if the system creates an image to communicate the idea of "poverty," do viewers of the image think of poverty or related concepts?) We can think of this as *semantic transferability* – the system's semantic model and perceptual grounding are similar enough to the viewer's semantic model and perceptual grounding to allow a conceptual transfer between the two.

Measuring a system's understanding and communication ability in a relative sense means asking whether the system's semantic model can be said to make sense (i.e., conceptual relationships in the model are somehow defensible). We can think of this as *semantic coherence* – the system's semantic model and perceptual grounding are justifiable, given its experience.

We can measure both of these, at least to some extent, with variations on human survey instruments and various clever uses of clustering techniques.

3.5.1 Evaluating Semantic Transferability

DARCI's ability to identify appropriate conceptual labels for images has been tested on various datasets, including:

- *IAPS* – 390 images labeled with scores for eight affective categories: *amusement*, *awe*, *contentment*, *excitement*, *anger*, *disgust*, *fear*, and *sadness* (Mikels et al., 2005, 4).
- *Art Photo* – 806 images of art that were labeled by their creator with one of the above eight affective categories (Machajdik & Hanbury, 2010).
- *Abstract* – 280 images of abstract paintings labeled by volunteers with one of the above eight affective categories (Machajdik & Hanbury, 2010).
- *DARCI* – approximately 15,000 images labeled by volunteers with a variable number of adjectives from a set of over 18,000 possibilities.

Of note, DARCI's single, general set of perceptual features and their associated appreciation networks (Section 3.2.1.1) perform well (Norton, Heath, & Ventura, 2017) on the first three datasets when compared with state-of-the-art affective classification algorithms (Machajdik & Hanbury, 2010; Yanulevskaya et al., 2008), even though those algorithms use specialized feature sets tailored for each data set and for each conceptual label. Additionally, DARCI understands many more (by an order of magnitude) than the eight concepts represented in those data sets. While this type of evaluation is necessary and fairly straightforward, it does not get to the heart of the matter. It is what DARCI does with this demonstrated proficiency that is important, and this is more difficult to evaluate.

If the system's goal is to communicate intent through, for example, visual metaphor or music, then at some level viewers of the metaphor or listeners to the music should agree on what that intention is. There is a natural way to measure this, in the form of human surveys, with their well-known strengths and weaknesses.

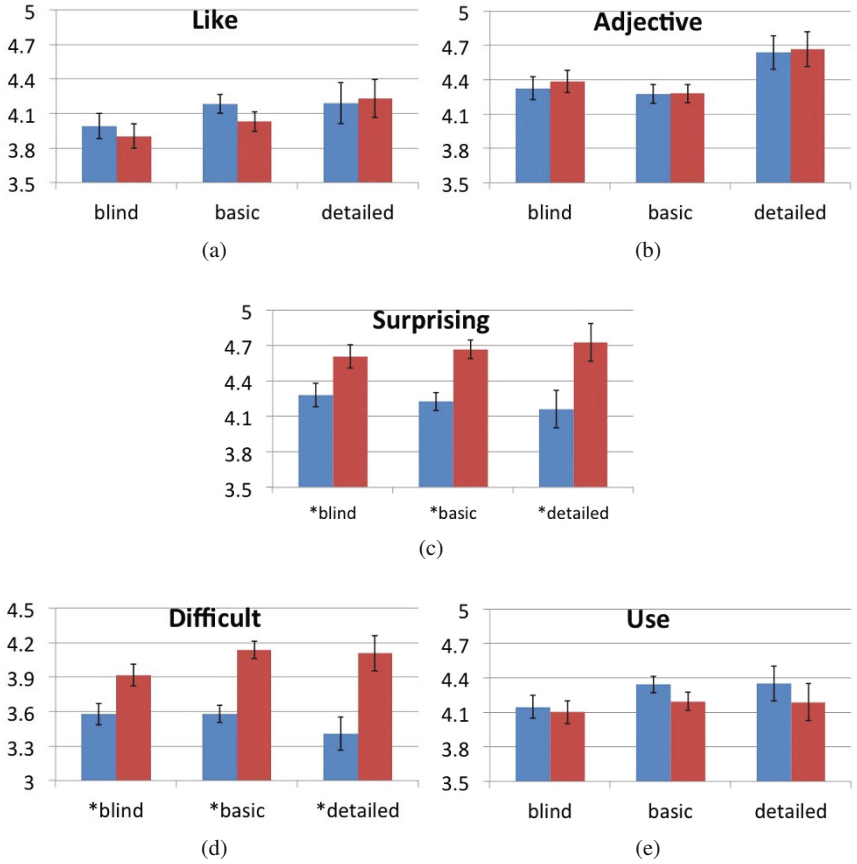
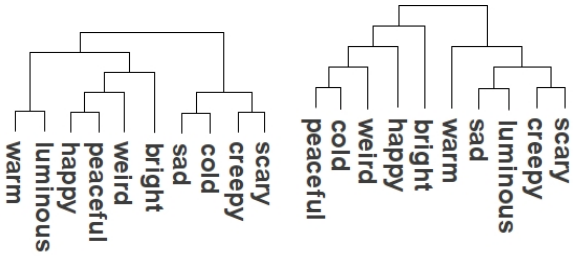


Fig. 3.8 Four human artists and DARCI created image renderings with the goal of communicating a simple concept (the human artists were limited to the use of filters similar to those available to DARCI). Survey participants were randomly given one of three levels of information about DARCI and then randomly shown images from those created and asked to evaluate them using a series of five Likert items. Here are shown the average scores (with standard error) for the five items [(a) – (e)] across all artefacts produced by either humans (blue) or DARCI (red) for three different levels of information given to participants (blind, basic, and detailed). An asterisk marks those items for which the difference in Likert scores is statistically significant (Norton, Heath, & Ventura, 2015).

Semantic transferability can be measured directly by questioning, for example, "what does this picture make you think of?" or "what does this music make you feel?" or "choose the most appropriate label, etc. (see Fig. 3.8); or it can be measured indirectly by comparing clustering characteristics between the semantic models of the system and observer, for example, does the model cluster music similarly to how listeners cluster music? (see Fig. 3.9).



(a) (b)

Fig. 3.9 (a) Results of an agglomerative clustering of 10 adjectives associated with images created by DARCI using a distance metric inferred from a human survey. Participants were asked to sort 10 randomly chosen images created by DARCI (with an associated concept adjective that was unknown to the participants), according to their visual/conceptual similarity. Averaging these sorting results over all participants gives an approximation to a conceptual distance between the images (and thus their associated adjectives) as perceived by the participants. (b) Results of an agglomerative clustering of 10 adjectives associated with images created by DARCI using a Euclidean distance function in image feature space, representing the conceptual distance between images as perceived by DARCI.

3.5.2 Evaluating Semantic Coherence

If we operate on the assumption that semantics is (at least partially) captured by proximity within some semantic space, then we can measure semantic coherence, at least indirectly, with various kinds of clustering in that space. Semantic coherence of a model (even one quite different from what we might naturally understand) can be measured by observing whether the clustering behavior of artefacts *perceived* by the system is similar to the clustering behavior of concepts *conceived* by the system (e.g., under the model, do pictures about war cluster closer to each other than they do to, say, pictures about eating). In other words, artefacts created by the system should cluster in ways that reflect the semantic similarity of the concepts on which they are based. For example, 'scary' and 'creepy' images should cluster together more closely (i.e., be harder to tell apart) than 'cold' and 'happy' images, because 'scary' and 'creepy' are more similar in meaning than 'cold' and 'happy'. By using clustering, we may not be able to objectively tell if a *specific* image conveys a *particular* adjective, but we can objectively see how well the system in general is creating images that reflect the relationships in its learned semantic model.

Perhaps the simplest way to do this is to cluster artefacts created by the system and look at the "quality" of the clusters. This can be done in a number of ways, for example using cluster entropy and purity metrics. In a good clustering, the entropy will be low and many clusters will have a high purity. Artefacts associated with semantically distinct concepts will be more easily clustered "correctly" than will those associated with semantically similar concepts that are more easily confused. In multiple experiments with DARCI this has been shown to be the case, both for

concepts the system has learned and for those it has not (Heath, Norton, & Ventura, 2014; Heath & Ventura, 2016).

Another way to evaluate such a clustering is to perform it agglomeratively and observe the clustering behavior as the number of clusters is reduced. Artefacts associated with semantically similar concepts should cluster together sooner than those associated with semantically distinct concepts. Fig. 3.10 shows an example of this for some images created by DARCI.

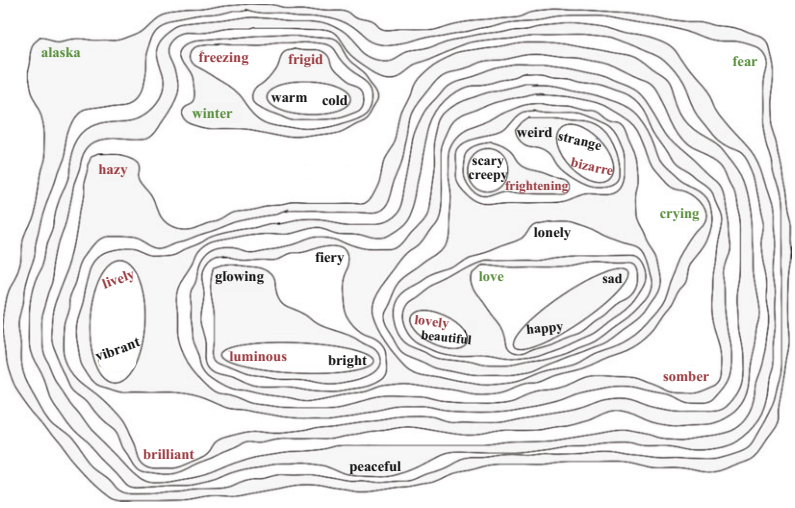
Note that perceptual differences between words may not always correspond to (what we initially think of as) their semantic differences. For example, in Fig. 3.10 we might expect the adjectives "warm" and "cold" to have distinct visual qualities that would significantly separate them in image space (Fig. 3.10b); after all, they are opposites! However, in Fig. 3.10a we see that "warm" and "cold" are semantically similar (at least relative to the other words), so DARCI's renderings of these adjectives can actually look similar. Though at first blush, this might seem unfortunate, it is actually another indication that DARCI is accurately generating images according to it's learned semantic relationships. Of course, one might argue that the learned model is wrong, but this can quickly devolve into a philosophical debate. One approach to this "misleading" semantics is to consider it as an opportunity to explore interesting kinds of slippage, jokes and various forms of dissonance.

In some cases, it is possible to demonstrate that such clustering techniques are consistent with human evaluators (Heath et al., 2014; Murray & Ventura, 2012), lending even more weight to automated clustering techniques as a viable method of evaluation that does not rely on human evaluation (this is not meant to dismiss the importance of human evaluation – critical in many areas of computational creativity – but to relax our overreliance on an approach that can be expensive, time-consuming and inconsistent). One way to demonstrate this consistency is simply to have humans perform the same clustering task and compare the two clusterings (as is shown in Fig. 3.9).

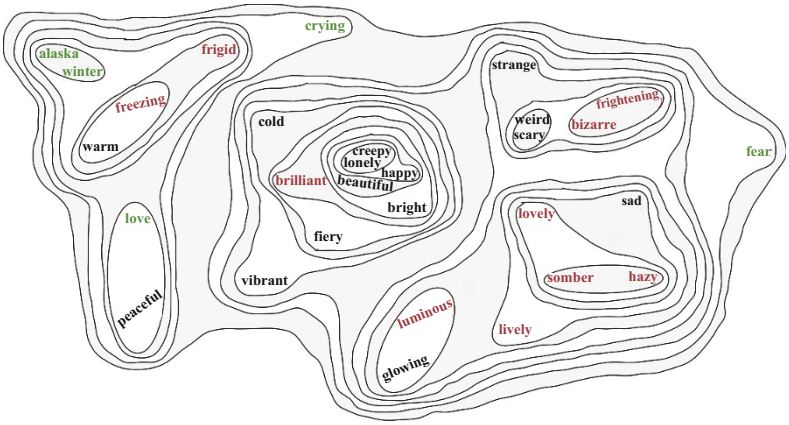
3.6 Concluding Remarks

We have argued that autonomy and intentionality are key characteristics for computationally creative systems and that while it is challenging to satisfactorily formalize them, there is still much that can be done both to instantiate and to evaluate these characteristics in the systems we build. To support this latter claim, we have offered examples from the DARCI system that demonstrate various aspects of autonomy and intentionality as well as results from several evaluation studies that validate this. However, even given the arguments presented here, we are only beginning to glimpse the real potential for computational creativity.

It is possible that, as with scientific theories, computational creativity may never be satisfactorily demonstrated, only controverted. If this is the case, then just as scientific theories are constructed to resist falsification, system builders must construct systems that resist demonstrations against their creative potential. It is in this spirit that we



(a) Vector Space (300 dimensions)



(b) Image Feature Space (51 dimensions)

Fig. 3.10 A 2D visualization of the spatial relationships between the words in (a) *vector space*, compared with the spatial relationships of their respective images in (b) *image feature space*. For the visualization in vector space, we used multidimensional scaling to find an approximate 2D plot (from 300 dimensions) of the distances between each word's vector. We then did agglomerative clustering (using expectation-maximization) with 30 word vectors and drew the resulting clusters on the 2D plot. For the visualization in image feature space, we calculated the average feature vector of 10 separately rendered images for each of the 30 words. We then performed multidimensional scaling (from 51 dimensions) and agglomerative clustering in the same way we did with the word vectors. Red words are adjectives on which DARCI was never trained, while green words are nonadjectives. Differences between the word clusters and the image clusters are to be expected as the visual semantic model learns from noisy data and multidimensional scaling has to approximate 2D positions from a high-dimensional space. Even so, the image clusters roughly correspond to the word clusters, demonstrating that DARCI is able to render images that are semantically cohesive, even for words on which DARCI was never trained, including nonadjectives.

address some of the most critical points of the debate – intentionality, autonomy and evaluation – and contend that it is possible to build systems that are resistant to efforts to prove that they lack these characteristics, though there is still a great deal of room for further sophistication, generality and robustness.

Another tactic for championing the idea of computational creativity is an educational one, in which we as a field engage with the public (Colton & Ventura, 2014; Norton, Heath, & Ventura, 2011a), exposing them to the idea of computational creativity, dispelling unfounded concerns and helping demonstrate the benefits we see in creative artificial systems.

A combination of these approaches, along with continuing advances in our own understanding and system building abilities will eventually result in a general acknowledgment that creativity is likely not within the purview of humanity only, nor the last bastion of differentiation between ourselves and the machines.

References

Colton, S. (2008). Creativity versus the perception of creativity in computational systems. In *Proceedings of the AAAI Spring Symposium on Creative Systems*.

Colton, S. (2011). The Painting Fool: Stories from building an automated painter. In J. McCormack & M. d'Inverno (Eds.), *Computers and Creativity*. Springer-Verlag.

Colton, S., & Ventura, D. (2014). You can't know my mind: A festival of Computational Creativity. In *Proceedings of the 5th International Conference on Computational Creativity* (pp. 351–354).

Csikszentmihalyi, M., & Robinson, R. E. (1990). *The art of seeing*. The J. Paul Getty Trust Office of Publications.

Csurka, G., Dance, C. R., Fan, L., Willamowski, J., & Bray, C. (2004). Visual categorization with bags of keypoints. In *Proceedings of the workshop on statistical learning in computer vision* (pp. 1–22).

Denoyer, L., & Gallinari, P. (2006). The Wikipedia XML corpus. In *INEX Workshop pre-proceedings* (pp. 367–372).

Heath, D., Norton, D., & Ventura, D. (2013). Autonomously communicating conceptual knowledge through visual art. In *Proceedings of the 4th International Conference on Computational Creativity* (pp. 97–104).

Heath, D., Norton, D., & Ventura, D. (2014). Conveying semantics through visual metaphor. *ACM Transactions on Intelligent Systems and Technology*, *5*(2), article 31.

Heath, D., & Ventura, D. (2016). Creating images by learning image semantics using vector space models. In *Proceedings of the Association for the Advancement of Artificial Intelligence* (pp. 1202–1208).

Johnson, D., & Ventura, D. (2016). Musical motif discovery from non-musical inspiration sources. *ACM Computers in Entertainment—Special Issue on Musical Metacreation*, *14*(2), article 7.

Kiss, G. R., Armstrong, C., Milroy, R., & Piper, J. (1973). An associative thesaurus of English and its computer analysis. In A. J. Aitkin, R. W. Bailey, & N. Hamilton-Smith (Eds.), *The computer and literary studies*. Edinburgh, UK: Edinburgh University Press.

Lebaron, D. M., Mitchell, L. A., & Ventura, D. (2015). Intelligent content generation via abstraction, evolution and reinforcement. In *Proceedings of the AIIDE Workshop on Experimental AI in Games* (pp. 36–41).

Machajdik, J., & Hanbury, A. (2010). Affective image classification using features inspired by psychology and art theory. In *Proceedings of the International Conference on Multimedia* (pp. 83–92).

McCorduck, P. (1991). *AARON's code: Meta-art, artificial intelligence, and the work of Harold Cohen*. W. H. Freeman & Co.

Mikels, J. A., Fredrickson, B. L., Larkin, G. R., Lindberg, C. M., Maglio, S. J., & Reuter-Lorenz, P. A. (2005). Emotional category data on images from the international affective picture system. *Behavior Research Methods*, *37*, 626–630.

Mikolov, T., Chen, K., Corrado, G., & Dean, J. (2013). Efficient estimation of word representations in vector space. In *Proceedings of the International Conference on Learning Representations*.

Morris, R., Burton, S., Bodily, P., & Ventura, D. (2012). Soup over bean of pure joy: Culinary ruminations of an artificial chef. In *Proceedings of the 3rd International Conference on Computational Creativity* (pp. 119–125).

Mumford, M., & Ventura, D. (2015). The man behind the curtain: Overcoming skepticism about creative computers. In *Proceedings of the 6th International Conference on Computational Creativity* (pp. 1–7).

Murray, S. J., & Ventura, D. (2012). Algorithmically flexible style composition through multi-objective fitness functions. In *Proceedings of the 1st International workshop on musical metacreation* (pp. 55–62).

Nelson, D. L., McEvoy, C. L., & Schreiber, T. A. (1998). The University of South Florida word association, rhyme, and word fragment norms. http://www.usf.edu/FreeAssociation/.

Norton, D., Heath, D., & Ventura, D. (2010). Establishing appreciation in a creative system. In *Proceedings of the 1st International Conference on Computational Creativity* (pp. 26–35).

Norton, D., Heath, D., & Ventura, D. (2011a). An artistic dialogue with 'the artificial. In *Proceedings of the 8th ACM Conference on Creativity and Cognition* (pp. 31–40).

Norton, D., Heath, D., & Ventura, D. (2011b). Autonomously creating quality images. In *Proceedings of the 2nd International Conference on Computational Creativity* (pp. 10–15).

Norton, D., Heath, D., & Ventura, D. (2013). Finding creativity in an artificial artist. *Journal of Creative Behavior*, *47*(2), 106–124.

Norton, D., Heath, D., & Ventura, D. (2015). Accounting for bias in the evaluation of creative computational systems: An assessment of DARCI. In *Proceedings of the 6th International Conference on Computational Creativity* (pp. 31–38).

Norton, D., Heath, D., & Ventura, D. (2017). Improving affective image annotation with features that summarize local interest points. *Submitted*.

Powers, D. M. W. (2011). Evaluation: From precision, recall and F-measure to ROC, informedness, markedness & correlation. *Journal of Machine Learning Technologies*, *2*, 37–63.

Sivic, J., Russell, B. C., Efros, A. A., Zisserman, A., & Freeman, W. T. (2005). Discovering objects and their location in images. *International Journal of Computer Vision*, *1*, 370–377.

Smith, M. R., Hintze, R. S., & Ventura, D. (2014). Nehovah: A neologism creator nomen ipsum. In *Proceedings of the 5th International Conference on Computational Creativity* (pp. 173–181).

Thomas, S., Boatman, E., Polyakov, S., Mumenthaler, J., & Wolff, C. (2013). The Noun Project. http://thenounproject.com.

Yanulevskaya, V., van Gemert, J. C., Roth, K., Herbold, A. K., Sebe, N., & Geusebroek, J. M. (2008). Emotional valence categorization using holistic image features. In *Proceedings of the 15th IEEE International Conference on Image Processing* (pp. 101–104).

Chapter 4
From Conceptual Mash-ups to Badass Blends: A Robust Computational Model of Conceptual Blending

Tony Veale

Abstract Conceptual blending is a complex cognitive phenomenon whose instances range from the humdrum to the pyrotechnical. Most remarkable of all is the ease with which we humans regularly understand and produce complex blends. While this facility will doubtless elude our best efforts at computational modeling for some time to come, there are practical forms of conceptual blending that are amenable to computational exploitation right now. In this chapter we introduce the notion of a *conceptual mash-up*, a robust form of blending that allows a computer to creatively reuse and extend its existing commonsense knowledge of a topic. We show also how a repository of such knowledge can be harvested automatically from the web, by targeting the casual questions that we pose to ourselves and to others every day. By acquiring its world knowledge from the questions of others, a computer can eventually learn to pose introspective questions of its own, in the service of its own creative *mash-ups*.

4.1 The Plumbing of Creative Thought

We can think of figurative comparisons as pipes that carry salient information from a source domain to a target domain. Some figurative pipes are thin, such as a simile that transfers just a single property from a source concept onto a target idea (Hao & Veale, 2010). Other pipes are fat, and so convey a good deal more information: think of the resonant metaphors (Veale, 2012; Veale, Shutova, & Klebanov, 2016) and evergreen analogies (Goel, 2019) that yield deeper meanings the more you look at them. By convention, most pipes carry information in one direction only, from the source domain to the target domain. But creativity is no respecter of convention, and creative comparisons are sometimes a two-way affair, in which aspects of the source and target are thoroughly mixed together in a back-and-forth exchange of ideas at the boundary of seemingly very different domains (Lavrač et al., 2019), to create

School of Computer Science, University College Dublin, Ireland. e-mail: tony.veale@gmail.com

T. Veale, F. A. Cardoso (eds.), *Computational Creativity*, Computational Synthesis and Creative Systems, https://doi.org/10.1007/978-3-319-43610-4_4

something utterly new and imaginative. To appreciate the differences between these different kinds of figurative plumbing, consider the following excerpt from the script for the movie Jurassic Park, which captures an exchange between the park's creator, John Hammond, and a wry mathematician, Ian Malcolm, who has been asked to evaluate the park's viability before it is opened to the public. The park of the title is populated with genetically engineered dinosaurs, so the dialogue takes place against a backdrop of carnivorous mayhem and rampant destruction:

> **John Hammond**: All major theme parks have delays. When they opened Disneyland in 1956, nothing worked!
>
> **Dr. Ian Malcolm**: Yeah, but, John, if The Pirates of the Caribbean breaks down, the pirates don't eat the tourists.

At this point in the movie, nothing is working in Jurassic Park, but nothing worked in 1956 at Disneyland either, and the latter turned out to be a huge financial and cultural success. Hammond thus frames Disneyland as a triumph, by focusing on the temporal sequence of events associated with its launch, its initial problems, and its eventual success. With this implicit analogy to Jurassic Park, whose launch has been plagued by unique problems of its own, Hammond predicts that his own troubled venture will follow the same script and achieve the same success. In effect, he sees Disneyland and Jurassic Park as two overlapping frames (much as in Lavrač et al. (2019)), and wants others to see the overlap too, so they might come to the same conclusions. Malcolm's rejoinder is also intended to be understood in the context of this analogy, but it is much more than an analogy as conceived in (Falkenhainer, Forbus, & Gentner, 1989; Gentner, 1983; Gentner, Falkenhainer, & Skorstad, 1987; Goel, 2019; Veale & Keane, 1997). It involves mapping, yes, so that The Pirates of the Caribbean is aligned with the attractions of Jurassic Park and the pirates of the former are mapped to the dinosaurs of the latter. But the salient behaviors of the latter, such as eating people willy-nilly, are also integrated with the protagonists of the former, to generate a counterfactual image of animatronic pirates eating tourists in mouse-earred caps. In the words of Fauconnier and Turner (1994, 2002), Malcolm has created a *blend* and is now *running the blend*: that is, he is conducting a mental simulation to explore the emergent possibilities that were hitherto just latent in the juxtaposition of both conceptual frames.

When the actor and writer Ethan Hawke was asked to write a profile of Kris Kristofferson for *Rolling Stone* magazine, Hawke had to create an imaginary star of his own to serve as an apt contemporary comparison. For Hawke, Brad Pitt is as meaningful a comparison as one can make, but even Pitt's star power is but a dim bulb to that of Kristofferson when he shone most brightly in the 1970s. To communicate just how impressive the singer-actor-activist would have seemed to an audience in 1979, Hawke assembled the following Frankenstein monster from the body of Pitt and other assorted star parts:

> Imagine if Brad Pitt had written a No. 1 single for Amy Winehouse, was considered among the finest songwriters of his generation, had been a Rhodes scholar, a U.S. Army Airborne Ranger, a boxer, a professional helicopter pilot – and was as politically outspoken as Sean Penn. That's what a motherfuckin' badass Kris Kristofferson was in 1979.

Pitt comes off poorly in the comparison, but this is precisely the point: no contemporary star comes off well, because in Hawke's view, none has the wattage that Kristofferson had in 1979. The awkwardness of the comparison, and the fancifulness of the blended image, serves as a creative meta-description of Kristofferson's achievements. In effect Hawke is saying, "look at what lengths I must go to find a fair comparison for this man without peer." Notice also how salient information flows in both directions in this comparison. To create a more rounded comparison, Hawke finds it necessary to mix in a few elements from other stars (such as Sean Penn), and to also burnish Pitt's résumé with elements borrowed from Kristofferson himself. Most of this additional structure is imported literally from the target, as when we are asked to imagine Pitt as a boxer or a helicopter pilot. Other structure is imported in the form of an analogy: while Kristofferson wrote songs for Janis Joplin, Pitt is imagined as a writer for her modern counterpart, Amy Winehouse.

This Pitt 2.0 does not actually exist, of course. Like Ian Malcolm's view of Jurassic Park qua Disneyland, Hawke's description is a conceptual blend that constructs a whole new source concept in its own counterfactual space. Blending is pervasive in modern culture, and can be seen in everything from cartoons to movies to popular fiction, while the elements of a blend can come from any domain of experience, from classic novels to 140-character tweets to individual words. As defined by Fauconnier (1994, 1997) and Fauconnier and Turner (1994, 2002), conceptual blending combines the smoothness of metaphor with the structural complexity and organizing power of analogy. We can think of blending as a cognitive operation in which conceptual ingredients do not flow in a single direction, but are thoroughly stirred together, to create a new structure with its own emergent meaning. Moreover, a blend can itself be used as a component part in larger blends, to create pyrotechnical flourishes of language that dazzle and amaze but rarely overtax our powers of conceptual analysis. Consider the following snarky comparison, freshly minted for Sam Mendes in the *Guardian* newspaper after studio bosses had chosen him to direct the 23rd film in the *James Bond* franchise: "Appearance: Like the painting in George Clooney's attic." This is not a simple comparison, but a complex blend that is loaded with figurative meaning, and we require neither a prior mental image of Sam Mendes nor a knowledge of the paintings in Clooney's attic to understand its real meaning. We can be quite certain that the picture in question is not a real picture that Clooney might actually own, whether *A Rake's Progress* or *Dogs Playing Poker*, but an entirely fictional painting that we create on the fly, via Fauconnier and Turner's process of conceptual blending. As Fauconnier and Turner might say, this is a multilayered blend that must be unpacked in several stages. The blend exploits our familiarity with Oscar Wilde's *The Picture of Dorian Gray*, a morality tale concerning the fate of a handsome but narcissistic young man who pledges his soul so that his painted self might suffer the ravages of time in his stead. Dorian soon discovers that his portrait – the infamous "painting in the attic" – not only changes to reflect his true age, but also holds a mirror to his inner being. As Dorian descends into moral degeneracy, his painted counterpart suffers a more physical degeneration, for as Dorian's portrait becomes increasingly ugly to behold he himself remains preternaturally youthful.

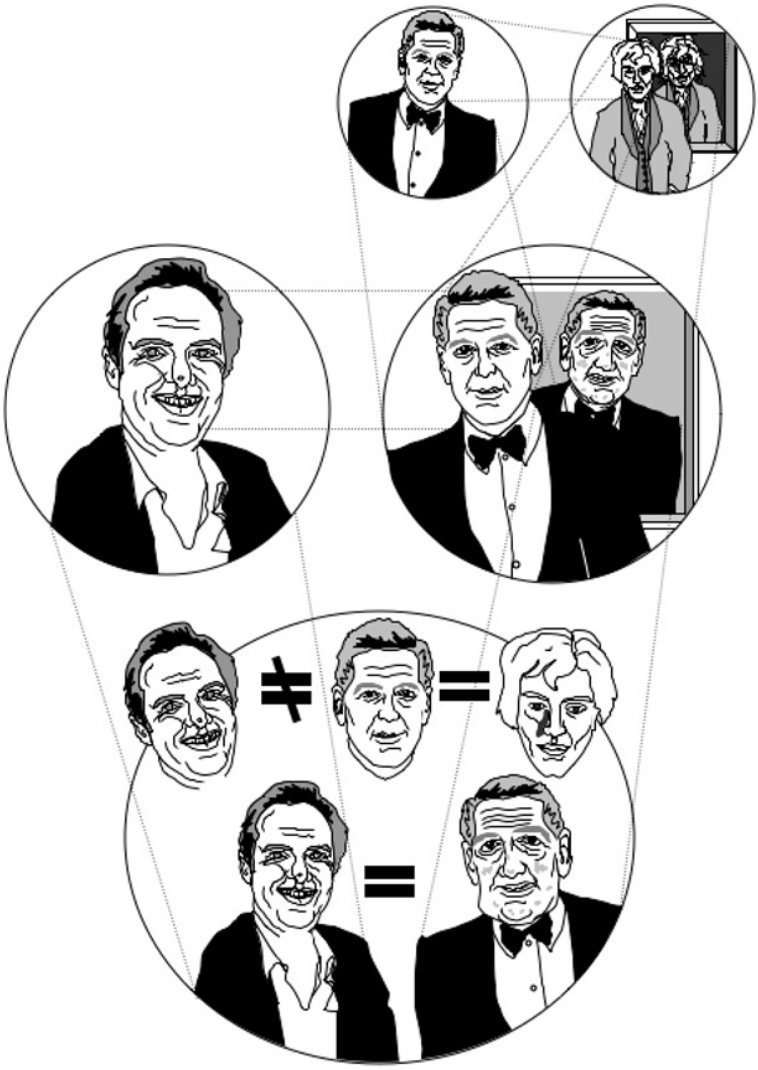


Fig. 4.1 The *blend-within-a-blend* that underpins the *Guardian*'s comparison of director Sam Mendes to "the picture in George Clooney's attic" (reproduced from Veale (2012)).

In the right hands, any cliché can be revitalized in a well-turned phrase, and the *Guardian* breathes humorous new life into the Dorian Gray cliché-archetype by embedding it within two more topical nested blends. A visual representation of the workings of this blend-within-a-blend is provided in Figure 4.1. The inner blend re-imagines Wilde's story with a new leading man, George Clooney, whose matinée-idol good looks make him an apt substitute for the handsome youth of the original tale. Clooney has maintained his status as a Hollywood sex symbol for

almost two decades, and he remains a regular fixture in the pages of celebrity gossip sheets. We find it easy to imagine a slowly decaying portrait in a dark corner of Clooney's attic, and even if the conceit has the tang of sour grapes, this just adds to its snarkily humorous effect. Note that this inner blend is more than a simile, a metaphor or an analogy, and it does more than compare George Clooney to Dorian Gray. Rather, it creates a new version of the morality tale with its very own star. In the world of the blend, Clooney really does have a portrait of his aging, sin-wracked face in the attic. This inner blend puts a new face on Wilde's tale, to create a new chunk from familiar elements that can now be reused in other blends, as though it had always existed in our cultural lexicons.

The *Mendes-as-Clooney-as-Dorian* and *Kristofferson-as-Pitt* blends show just how complex a blend can be, while nonetheless remaining intelligible to a reader: when we interpret these constructs, we are not aware of any special challenge being posed, or of any special machinery being engaged. Nonetheless, this kind of blend poses significant problems for our computers and their current cognitive/linguistic modeling abilities. So in this chapter we present a computational middle ground, called a conceptual mash-up, that captures some of the power and utility of a conceptual blend, but in a form that is practical and robust to implement on a computer. From this starting point we can begin to make progress toward the larger goal of creative computational systems that —- to use Hawke's word —- can formulate truly badass blends of their own.

Creative language is a knowledge-hungry phenomenon. We need knowledge to create or comprehend an analogy, metaphor, or blend, while these constructs allow us to stretch our knowledge into new forms and niches. But computers cannot be creative with language unless they first have something that is worth saying creatively, for what use is a poetic voice if one has no opinions or beliefs of one's own that need to be expressed? This current work describes a reusable resource — a combination of knowledge and of tools for using that knowledge —- that can allow other computational systems to form their own novel hypotheses from mash-ups of common stereotypical beliefs. These hypotheses can be validated in a variety of ways, such as via a web search, and then expressed in a concise and perhaps creative linguistic form, such as in poem, metaphor, or riddle. The resource, which is available as a public web service called *Metaphor-Eyes*, produces conceptual mash-ups for its input concepts, and returns the resulting knowledge structures in an XML format that can then be used by other computational systems in a modular, distributed fashion. The *Metaphor-Eyes* service is based on an approach to creative introspection first presented in Veale and Li (2011), in which stereotypical beliefs about everyday concepts are acquired from the web, and then blended on demand to create hypotheses about topics that the computer may know little or nothing about. We present the main aspects of *Metaphor-Eyes* in the following sections, and show how its capacity for conceptual mash-ups can be exploited by other systems via the web.

4.1.1 Structure of This Chapter

Our journey begins in the next section, with a brief overview of relevant computational work in the areas of metaphor and blending. It is our goal to avoid hand-crafted representations, so in the section after that we describe how the system can acquire its own commonsense knowledge from the web, by eavesdropping on the revealing questions that users pose every day to a search engine such as Google. This knowledge provides the basis for conceptual mash-ups, which are constructed by repurposing web questions to form new introspective hypotheses about a topic. We also introduce the notion of a *multisource mash-up*, which allows us to side-step the vexing problem of context and user intent in the construction of conceptual blends. Finally, an empirical evaluation of these ideas is presented, and the chapter concludes with thoughts on future directions.

4.2 Related Work and Ideas

We use metaphors and blends not just as rhetorical flourishes, but as a basis for extending our inferential powers into new domains (Barnden, 2006). Indeed, work on analogical metaphors shows how metaphor and analogy use knowledge to create knowledge (Veale et al., 2016). *Structure-Mapping Theory*, or SMT (Gentner, 1983; Gentner et al., 1987), argues that analogies allow us to impose structure on a poorly understood domain, by mapping knowledge from one that is better understood. SME, the *Structure-Mapping Engine* (Falkenhainer et al., 1989) implements these ideas by identifying subgraph isomorphisms between two mental representations (see also Veale and Keane (1997) for a discussion of why this task is NP-hard). SME then projects connected substructures from the source to the target domain. SMT prizes analogies that are systematic in their preservation of causal structure (see also Winston (1980) and Carbonell (1982)), yet a key issue in any structural approach is how a computer can acquire structured representations for itself.

Veale and O'Donoghue (2000) proposed an SMT-based model of conceptual blending that was perhaps the first computational model of the phenomenon. The model, called *Sapper*, addresses many of the problems faced by SME —- such as deciding for itself which knowledge is relevant to a blend —- but succumbs to others, such as the need for a hand-crafted knowledge base. Pereira (2007) and Martins, Pereira, and Cardoso (2019) present an alternative computational model that combines SMT with other computational techniques, such as using genetic algorithms to search the space of possible blends. Pereira's model was applied both to linguistic problems (such as the interpretation of novel noun-noun compounds) and to visual problems, such as the generation of novel monsters/creatures for video games. Nonetheless, Pereira's approach was just as reliant on hand-crafted knowledge. To explore the computational uses of blending without such a reliance on specially crafted knowledge, Veale (2006) showed how blending theory can be used to understand novel portmanteau words — or "formal" blends — such as

"Feminazi" (*Femi*nist + *Nazi*). This approach, called *Zeitgeist*, automatically harvested and interpreted portmanteau blends from Wikipedia, using only the topology of Wikipedia itself and the contents of WordNet (Fellbaum, 1998) as resources.

The availability of large corpora and the web suggests a means of relieving the knowledge bottleneck that afflicts computational models of metaphor, analogy and blending. Turney and Littman (2005) showed how a statistical model of relational similarity can be constructed from web texts for handling proportional analogies of the kind used in SAT and GRE tests. No hand-coded or explicit knowledge is employed, yet Turney and Littman's system achieves an average human grade on a set of 376 SAT analogies (such as *mercenary:soldier::?:?*, where the best answer among four alternatives is *hack:reporter*). Almuhareb and Poesio (2004) described how attributes and values can be harvested for lexicalized concepts from the web, showing how these properties allow lexical concepts to be clustered into category structures that replicate the semantic divisions made by a curated resource such as WordNet (Fellbaum, 1998). Veale and Hao (2007a, 2007b, 2008) described how stereotypical knowledge can be acquired from the web by harvesting similes of the form "as P as C" (as in "as *smooth* as *silk*"), and went on to show, in Veale (2012), how a body of 4000 stereotypes could be used in a web-based model of metaphor generation and comprehension.

Shutova, Sun, and Korhonen (2010) combined elements of several of these approaches. They annotated verbal metaphors in corpora (such as "to stir excitement", where the verb "stir" is used metaphorically) with the corresponding conceptual metaphors identified by Lakoff and Johnson (1980) and listed in Lakoff, Espenson, and Schwartz (1991). Statistical clustering techniques were then used to generalize from the annotated exemplars, allowing their system to recognize other metaphors in the same vein (for example, "he swallowed his anger"). These clusters can also be analyzed to suggest literal paraphrases for a given metaphor (such as "to provoke excitement" or "suppress anger"). This approach is noteworthy for the way it operates with Lakoff and Johnson's inventory of conceptual metaphors without actually using any explicit knowledge of its own.

The questions people ask, and the web queries they pose, are an implicit source of commonsense knowledge. The challenge we face as computationalists lies in turning this *implicit* world knowledge into *explicit* representations. For instance, Pasca and Durme (2007) showed how knowledge of classes and their attributes can be extracted from the queries that are processed and logged by web search engines. We intend to show in this chapter how a commonsense representation that is derived from web questions can be used in a model of conceptual blending. We focus on well-formed questions, found either in the query logs of a search engine or harvested from documents on the web. These questions can be viewed as atomic properties of their topics, but they can also be parsed to yield logical forms for reasoning. We show here how we might, by representing topics via the questions we ask about them, also grow our knowledge base via blending, by posing these questions introspectively of other topics as well.

4.3 "Milking" Knowledge from the Web

Amid the ferment and noise of the World Wide Web sit nuggets of stereotypical world knowledge, in forms that can be automatically harvested. To acquire a property *P* for a topic *T*, one can look for explicit declarations of *T*'s *P-ness*, but such declarations are rare, as speakers are loathe to explicitly articulate truths that are tacitly assumed by others. Hearst (1992) observed that the best way to capture tacit truths in large corpora (or on the web) is to look for stable linguistic constructions that presuppose the desired knowledge. So rather than look for "*all Xs are Ys*", which is a laudably direct but exceedingly rare pattern in everyday usage, more frequent Hearst-patterns such as "Xs and other Ys" presuppose exactly the same hypernymic relations. By mining presuppositions rather than declarations, a harvester can cut through the layers of noise and misdirection that are endemic to the web.

If W is a count noun denoting a topic T_W, then the query "why do W_{plural} *" (where "*" denotes a wildcard) allows us to retrieve questions posed about T_W on the web, in this case via the Google API. (If W is a mass noun or a proper name, we can instead use the query "why does W * ?") These two formulations show the benefits of using questions as extraction patterns: a query is framed by an opening WH-question word and a closing question mark, ensuring that a complete statement is retrieved (Google snippets often contain sentence fragments); and number agreement between "do"/"does" and W suggests that the question is syntactically well-formed (good grammar helps discriminate well-formed musings from random noise). Queries with the subject T_W are dispatched whenever the system wishes to learn about a topic T. We ask the Google API to return 200 snippets per query, which are then parsed to extract well-formed questions and their logical forms. Questions that cannot be parsed in this way are rejected as being too complex for later reuse in conceptual blending.

For instance, the topic *Pirate* yields the query "why do pirates *", which can be used to retrieve snippets about pirates. The retrieval set includes these questions:

Why do pirates wear eye patches?
Why do pirates hijack vessels?
Why do pirates have wooden legs?

Parsing the second question above, we obtain its logical form:

$$\forall\, x\ pirate(x) \rightarrow \exists\, y\ vessel(y) \wedge hijack(x,y).$$

A computational system needs a critical mass of such commonsense knowledge before it can be usefully applied to problems such as conceptual blending. Ideally, we could extract a large body of everyday musings from the query logs of a search engine like Google, since many users persist in using full natural language questions as web queries. Yet such logs are jealously guarded, not least on concerns about privacy. Nonetheless, engines like Google do expose the most common queries in the form of text completions: as one types a query into the search box, Google anticipates

the user's query by matching it against past queries, and offers a variety of popular completions. These completions are a rich source of knowledge for a machine.

In an approach we call Google "milking," we coax completions from the Google search box for a long list of strings with the prefix "why do," such as "why do *a*" (which prompts "why do *a*nimals hibernate?"), and "why do *aa*' (which prompts "why do *aa* batteries leak?"). We use a manual trie-driven approach, using the input "why do *X*" to determine if any completions are available for a topic prefixed with *X*, before then drilling deeper with "why do *Xa*," ... "why do *Xz*," Though laborious, this process taps into a veritable mother lode of nuggets of conventional wisdom. Two weeks of milking yields approximately 25,000 of the most common questions on the web, for over 2000 topics, providing critical mass for the processes to come.

4.4 Conceptual "Mash-ups"

Google milking yields these frequent "Why do ..." questions about poets:

Why do poets repeat words?
Why do poets use metaphors?
Why do poets use alliteration?
Why do poets use rhyme?
Why do poets use repetition?
Why do poets write poetry?
Why do poets write about love?

Querying the web directly, the system finds other common presuppositions about poets, such as "why do poets die poor?" and "why do poets die young?', precisely the kind of knowledge that shapes our stereotypical view of poets yet which one is unlikely to see reflected in a dictionary's entries. Suppose a user asks the system to explore the ramifications of the blend *Philosophers are Poets*: this prompts the system to introspectively ask "how are philosophers like poets?". This question spawns others, which are produced by replacing the subject of the poet-specific questions above, yielding new introspective musings such as "do philosophers write poetry?", "do philosophers use metaphors?", and "do philosophers write about love?"

Each repurposed question can be answered by again appealing to the web: the system simply looks for evidence that the hypothesis in question (such as "philosophers use metaphors") is attested by literal usage in one or more web texts. The Google API finds supporting matches for the following hypotheses: "philosophers die poor" (3 hits), "philosophers die young" (6 hits), "philosophers use metaphors" (156 hits), and "philosophers write about love" (just 2 hits). The goal is not to show that these behaviors are as salient for philosophers as for poets, merely that they are attested to be meaningful for philosophers too. We refer to the construct *Philosophers are Poets* as a *conceptual mash-up*, since knowledge about a source concept, *Poet*, has been mashed-up with that of a target idea, *Philosopher*, to yield a new knowledge network for the latter. Conceptual mash-ups are a specific kind of conceptual blend, one that is easily constructed via simple computational processes.

To generate a mash-up, the system starts from a given target idea T and searches for the source concepts $S_1, \ldots, S_n$ that might plausibly yield a meaningful blend. A locality assumption limits the scale of the search space for $S_1, \ldots, S_n$, by assuming that T must exhibit a pragmatic similarity to any source concept S_i. Budanitsky and Hirst (2006) described a raft of term-similarity measures based on WordNet (Fellbaum, 1998), but what is needed for blending is a generative measure: one that can quantify the similarity of T to S as well as suggest a range of likely S_i's for any given topic T. We construct such a measure via corpus analysis, since a measure trained on corpora can easily be made corpus-specific and thus domain- or context-specific. The Google n-grams (Brants & Franz, 2006) provide a large collection of word sequences from web texts. Looking to the 3-grams, we extract coordinations of generic nouns of the form "Xs and Ys." For each coordination, such as "tables and chairs" or "artists and scientists," X is considered a pragmatic (rather than semantic) neighbor of Y, and vice versa. When identifying blend sources for a topic T, we consider the neighbors of T as candidate sources for a blend. Furthermore, if we consider the neighbors of T to be features of T, then a vector space representation for topics can be constructed, such that the vector for a topic T contains all of the neighbors of T that are identified in the Google 3-grams. This vector representation allows us to calculate the similarity of a topic T to a source S, and rank the neighbors $S_1, \ldots, S_n$ of T by their similarity to T (Veale & Li, 2013).

Intuitively, writers use the pattern "Xs and Ys" to denote an ad hoc category, so that the topics linked by this pattern are not just similar but truly comparable, or even interchangeable. Potential sources for T are ranked by their perceived similarity to T, as described above. So if one is generating mash-ups for *Philosopher*, the top-ranked sources found in the Google 3-grams are *Scholar*, *Epistemologist*, *Ethicist*, *Moralist*, *Naturalist*, *Scientist*, *Doctor*, *Pundit*, *Savant*, *Explorer*, *Intellectual* and *Lover*.

4.4.1 Multisource Mash-ups

The problem of finding good sources for a topic T is highly underconstrained, and depends on the contextual goals of the speaker. However, when blending is used for knowledge acquisition, multisource mash-ups allow us to blend a range of sources into a rich, context-free structure. If $S_1, \ldots, S_n$ are the n closest neighbors of T as ranked by similarity to T, then a mash-up can be constructed to describe the semantic potential of T by collating all of the questions from which the system derives its knowledge of $S_1, \ldots, S_n$, and by repurposing each question for T. A complete mash-up collates questions from all the neighbors of a topic, while a 10-neighbor mash-up for *Philosopher*, say, would collate all the questions associated with the top 10 neighbors *Scholar*, …, *Explorer* and insert "philosopher" as the subject of each. In this way a conceptual picture of *Philosopher* could be created, by drawing on beliefs such as that naturalists tend to be pessimistic and humanists care about morality.

A 20-neighbor mash-up for *Philosopher* would also integrate the system's knowledge of *Politician* into this picture, to suggest the possibilities that. for example, *philosophers lie*, *philosophers cheat*, *philosophers equivocate* and even that *philosophers have affairs* and *philosophers kiss babies*. Each of these hypotheses can be put to the test in the form of a specific web query; thus, the hypotheses "philosophers lie" (586 Google hits), "philosophers cheat" (50 hits), and "philosophers equivocate" (11 hits) are all validated with Google queries, whereas "philosophers kiss babies" (0 hits) and "philosophers have affairs" (0 hits) are not. As one might expect, the most domain-general hypotheses show the greatest promise of taking root in a target domain. For example, "why do artists use Macs?" is more likely to be successfully transplanted into another domain than "why do artists use perspective drawing?"

The generality of a question is related to the number of times it appears in our knowledge base with different subjects. Thus, "why do $< Xs >$ wear black" appears 21 times, while "why do $< Xs >$ wear black hats" and "why do $< Xs >$ wear white coats" each just appear twice. When a mash-up for a topic T is presented to the user, each imported question Q is ranked according to two criteria: Q_{count}, the number of neighbors of T that suggested Q as a hypothesis for T; and Q_{sim}, the similarity of T to its most similar neighbor that suggested Q (as calculated using a WordNet-based metric; see Budanitsky and Hirst (2006) for a survey; we use the metric of Veale and Li (2013) here). The two combine to give the single salience measure $Q_{salience}$, which is defined as follows:

$$Q_{salience} = Q_{sim} \times Q_{count} / (Q_{count} + 1).$$

Note that Q_{count} is always greater than 0, since each question Q must be suggested by at least one neighbor of T. Note also that salience is a measure of expectedness (see Grace and Maher (2019)) and thus of plausibility too, so when Q_{count} is large then so is $Q_{salience}$. It is time-consuming to dispatch every question in a mash-up to the web, as a mash-up of m questions requires m web queries. It is more practical to choose a cutoff n and simply test the top n questions, as ranked by $Q_{salience}$. In the next section we evaluate the ranking of questions in a mash-up, and estimate the likelihood of successful knowledge transfer from one topic to another.

4.5 Empirical Evaluation

Our corpus-attested, neighborhood-based approach to similarity does not use WordNet (Fellbaum, 1998), but is capable of replicating the same semantic divisions made by WordNet. In earlier work, Almuhareb and Poesio (2004) extracted features for concepts from text patterns instantiated on the web. Those authors tested the efficacy of the extracted features by using them to cluster 214 words taken from 13 semantic categories in WordNet (henceforth, we denote this experimental setup as AP214), and reported a cluster purity of 0.85 in replicating the category structures of WordNet. But if the neighbors of a term are instead used as features for that term, and if a

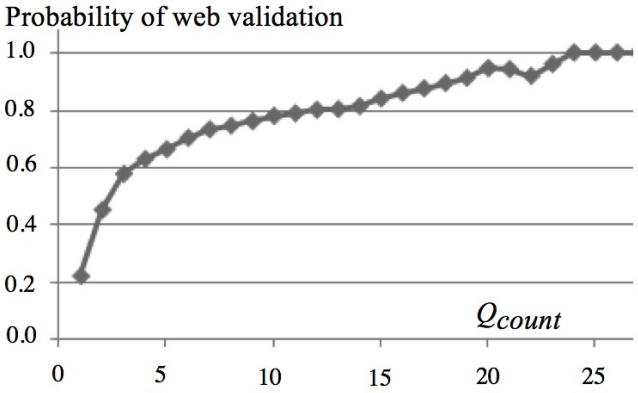


Fig. 4.2 Likelihood of a hypothesis in a mash-up being validated via web search (Y-axis) for hypotheses that are suggested by Q_{count} neighbors (X-axis).

term is also considered to be its own neighbor, then an even higher purity/accuracy of 0.934 is achieved on AP214. Using neighbors as features in this way requires a vector space of just 8300 features for AP214, whereas Almuhareb and Poesio's original approach to AP214 used approximately 60,000 features.

Just as knowledge tends to cluster into pragmatic neighborhoods, hypotheses likewise tend to be validated in clusters. As shown in Figure 4.2, the probability that a hypothesis is valid for a topic T grows with the number of neighbors of T for which it is known to be valid (that is, Q_{count}). Unsurprisingly, the closest neighbors with the highest similarity to the topic exert the most influence. Figure 4.3 shows that the probability of a hypothesis for a topic being validated by attested web usage grows with the number of the topic's neighbors that suggest it and its similarity to the closest of these neighbors (that is, $Q_{salience}$). In absolute terms, hypotheses perceived to have high salience (e.g., $> .6$) are much less frequent than those with lower ratings. So a more revealing test is the ability of the system to rank the hypotheses in a mash-up so that the top-ranked hypotheses have the greatest likelihood of being validated on the web. That is, to avoid information overload, the system should be able to distinguish the most plausible hypotheses from the least plausible, just as search engines like Google are judged on their ability to push the most relevant hits to the top of their rankings.

Figure 4.4 shows the average rate of web validation for the top-ranked hypotheses (ranked by salience) of complete mash-ups generated for each of our 10 test terms from all of their neighbors. Since these are common terms, they have many neighbors that suggest many hypotheses. On average, 85% of the top 20 hypotheses in each mash-up are validated by web search as plausible, while just 1 in 4 of the top 60 hypotheses in a mash-up are not validated by attested usage in web documents. Figures 4.2–4.4 show that the system is capable of acquiring knowledge from the web that can be successfully transferred to neighboring terms via metaphors and mash-ups, and then meaningfully ranked by salience. But just how useful is this knowledge? To

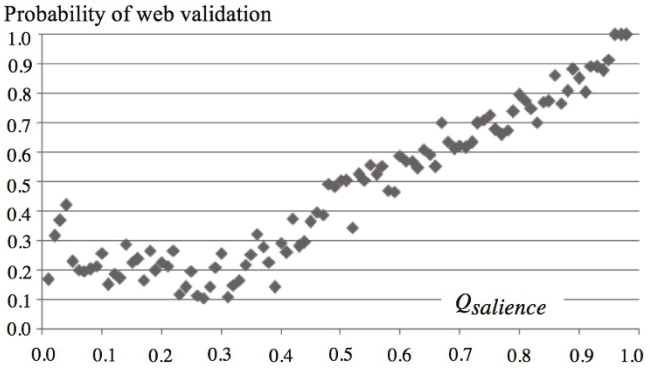


Fig. 4.3 Likelihood of a hypothesis in a mash-up being validated via web search (Y-axis) for hypotheses with a particular $Q_{salience}$ measure (X-axis).

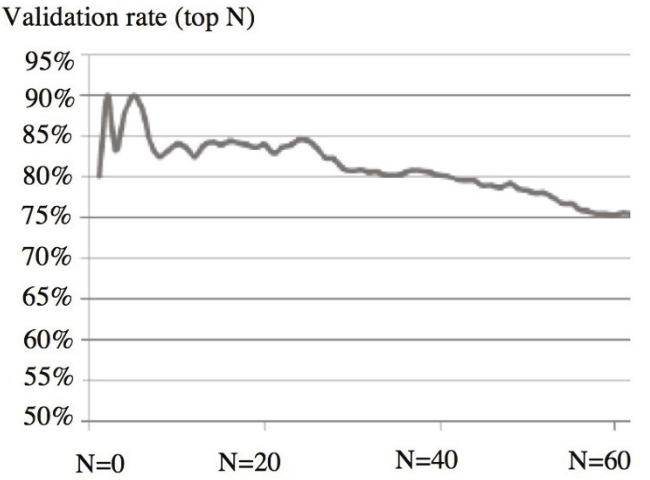


Fig. 4.4 Average percentage of the top N hypotheses in a mash-up (as ranked by $Q_{salience}$) that are validated by web search.

determine if it is the kind of knowledge that is useful for categorization, and thus the kind that captures the essence of a concept, we use it to replicate the AP214 test of Almuhareb and Poesio (2004). Recall that AP214 tests the ability of a feature set to support the category distinctions imposed by WordNet, so that the 214 words can be clustered back into the 13 WordNet categories from whence they came.

So, for each of these 214 words, we harvest questions from the web, and treat each question body as an atomic feature of its subject; thus, for example, we treat "kisses babies" as a feature of *Politician*. Clustering over these features alone offers poor accuracy when reconstructing WordNet categories, yielding a cluster purity of just over 0.5. One AP214 category in particular, comprising time units such as

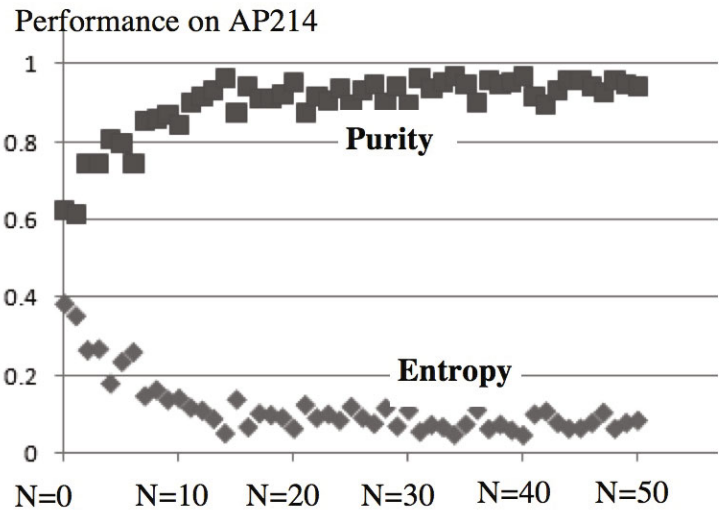


Fig. 4.5 Clustering performance on AP214 improves (that is, purity is higher and entropy is lower) as knowledge is transferred from the *N* closest neighbors of a term.

week and *year*, offers no traction to the question-based approach, and accuracy/purity increases to 0.6 when this category is excluded. People, it seems, rarely question the conceptual status of an abstract temporal unit on the web. Yet as knowledge is gradually transferred to the terms in AP214 from their corpus-attested neighbors, so that each term is represented as a conceptual mash-up of its *n* nearest neighbors, categorization markedly improves. Figure 4.5 demonstrates the increasing accuracy of the system on AP214 (excluding the vexing *time* category) when mash-ups of increasing numbers of neighbors are used. Blends really do bolster our knowledge of a topic with insights that are relevant to categorization.

4.6 Conclusions

We have explored how the most common questions on the web can provide the world knowledge needed to drive a robust, if limited, form of blending called a conceptual mash-up. The ensuing powers of self-questioning introspection, though basic, can be used to speculate upon the conceptual makeup of any given topic, not only in individual metaphors but also in rich, informative mash-ups of multiple concepts. The World Wide Web is central to this approach: not only are questions harvested from the web (for instance, via Google "milking"), but newly formed hypotheses are validated by means of simple web queries. The approach is practical, robust, and quantifiable, and uses an explicit knowledge representation that can be acquired on demand for a given topic. Most importantly, the approach makes a virtue of blending,

Fig. 4.6 A screenshot from the computational system *Metaphor-Eyes*, which implements the model described in this chapter. *Metaphor-Eyes* shows how we can use conceptual mash-ups to explore counterfactual and/or hybrid ideas and thus stimulate human creativity. (Note: Because the system has no prior ontological knowledge about Google, each entry above shows a default score of 100 and a support/similarity measure of 0.) Visit http://Afflatus.UCD.ie to interact with the *Metaphor-Eyes* system for yourself, or to find out more about the system's XML functionality.

and argues that we and our machines should view blending not just as a problem to be solved, but as a tool of creative computational engineering.

The ideas described here have been computationally realized in a web application and web service called *Metaphor-Eyes*. Figure 4.6 provides a snapshot of the service in action. The user enters a query —- in this case the provocative assertion "Google is a cult" — and the service provides an interpretation based on a mash-up of its knowledge of the source concept (*cults*) and of the target concept (*Google*). Two kinds of knowledge are used to provide the interpretation shown in Figure 4.6. The first is commonsense knowledge of cults, of the kind that we expect most adults to possess. This knowledge includes widely held stereotypical beliefs such as that cults are led by gurus, that they worship gods and enforce beliefs, and that they recruit new members, especially celebrities, who often act as apologists for the cult. The system possesses no stereotypical beliefs about Google, but using the Google 2-grams (somewhat ironically, in this case), it can find linguistic evidence for the notions of a "Google guru," a "Google god," and a "Google apologist." The

corresponding stereotypical beliefs about cults are then projected into the new blend space of *Google-as-a-cult*.

Metaphor-Eyes derives a certain robustness from its somewhat superficial treatment of blends as mash-ups. In essence, the system manipulates conceptual level objects (ideas and other blends) by using language level objects (strings, phrases, collocations) as proxies: a combination at the concept-level is deemed to make sense if a corresponding combination at the language-level can be found in a web corpus (or in the Google N-grams). As such, any creativity exhibited by the system is often facile or glib. Because the system looks for conceptual novelty in the veneer of surface language, it follows in the path of humor systems that attempt to generate interesting semantic phenomena by operating at the level of words and their conventional significations (see Gatti, Ozbal, Guerini, Stock, and Strapparava (2019) for other work in this vein).

We have thus delivered on just one half of the promise of our title. While conceptual mash-ups are something a computer can handle with relative ease, "badass" blends of the kind discussed in the introduction still lie far beyond our computational reach. Nonetheless, we believe the former provides a solid foundation for development of the tools and techniques that are needed to achieve the latter. Several areas of future research suggest themselves in this regard, and one that appears most promising at present is the use of mash-ups in the generation of poetry (see Veale (2013) for work in this direction). The tight integration of surface form and meaning that is expected in poetry means this is a domain in which a computer can serendipitously allow itself to be guided by the possibilities of word combination while simultaneously exploring the corresponding idea combinations at a deeper level (see Gervás (2019) for an exploration of key issues in computational poetry generation). Indeed, the superficiality of mash-ups makes them ideally suited to the surface-driven exploration of deeper levels of meaning.

Metaphor-Eyes should thus be seen as a community resource through which the basic powers of creative introspection (as first described in Veale and Li (2011)) can be made available to a wide variety of third-party computational systems. In this regard, *Metaphor-Eyes* is a single instance of what will hopefully become an established trend in the maturing field of computational creativity: the commonplace sharing of resources and tools, perhaps as a distributed network of creative web services (Veale, 2014), that will promote a wider cross-fertilization of ideas in our field. The integration of diverse services and components will in turn facilitate the construction of systems with an array of creative qualities. Only by pooling resources in this way can we hope to go beyond one-note systems and produce the impressive multinote "badass blends" of the title.

Acknowledgements This research was in part supported by the EC project WHIM, *The What-If Machine*, and by the EC project PROSECCO, a coordination action funded to *PROmote the Scientific Exploration of Computational Creativity.*

References

Almuhareb, A., & Poesio, M. (2004). Attribute-Based and Value-Based Clustering: An Evaluation. In *Proceedings of the 2004 conference on empirical methods in natural language processing (EMNLP)* (pp. 158–165). Association for Computational Linguistics.

Barnden, J. A. (2006). Artificial Intelligence, figurative language and cognitive linguistics. In *Cognitive linguistics: Current applications and future perspectives* (pp. 431–459). Berlin: Mouton de Gruyter.

Brants, T., & Franz, A. (2006). *Web 1T 5-gram Version 1*. Linguistic Data Consortium.

Budanitsky, A., & Hirst, G. (2006). Evaluating WordNet-based Measures of Lexical Semantic Relatedness. *Computational Linguistics*, *32*(1), 13–47.

Carbonell, J. (1982). Metaphor: An Inescapable Phenomenon in Natural Language Comprehension. In W. Lehnert & M. Ringle (Eds.), *Strategies for natural language processing* (pp. 415–434). Lawrence Erlbaum.

Falkenhainer, B., Forbus, K. D., & Gentner, D. (1989). The Structure Mapping Engine: Algorithm and Examples. *Artificial Intelligence*, *41*, 1–63.

Fauconnier, G. (1994). *Mental spaces: aspects of meaning construction in natural language*. Cambridge University Press.

Fauconnier, G. (1997). *Mappings in Thought and Language*. Cambridge University Press.

Fauconnier, G., & Turner, M. (1994). *Conceptual Projection and Middle Spaces (Technical report 9401)*. University of California at San Diego, Department of Computer Science.

Fauconnier, G., & Turner, M. (2002). *The Way We Think: Conceptual Blending and the Mind's Hidden Complexities*. New York: Basic Books.

Fellbaum, C. (Ed.). (1998). *Wordnet: An electronic lexical database (ISBN = 0-262-06197-x)* (First ed.). MIT Press.

Gatti, L., Ozbal, G., Guerini, M., Stock, O., & Strapparava, C. (2019). Computer-supported human creativity and human-supported computer creativity. In T. Veale & F. A. Cardoso (Eds.), *Computational creativity: The philosophy and engineering of autonomously creative systems* (pp. 235–252). Springer.

Gentner, D. (1983). Structure-mapping: A Theoretical Framework. *Cognitive Science*, *7*(2), 155–170.

Gentner, D., Falkenhainer, B., & Skorstad, J. (1987). Metaphor: The good, the bad and the ugly. In Y. Wilks (Ed.), *Proceedings of the 1987 workshop on theoretical issues in NLP* (pp. 176–180). Association for Computational Linguistics. Hillsdale, NJ.: Lawrence Erlbaum Associates.

Gervás, P. (2019). Exploring quantitative evaluations of the creativity of automatic poets. In T. Veale & F. A. Cardoso (Eds.), *Computational creativity: The philosophy and engineering of autonomously creative systems* (pp. 273–302). Springer.

Goel, A. (2019). Revisiting design, analogy, and creativity. In T. Veale & F. A. Cardoso (Eds.), *Computational creativity: The philosophy and engineering of autonomously creative systems* (pp. 139–156). Springer.

Grace, K., & Maher, M. L. (2019). Expectation-based models of novelty for evaluating computational creativity. In T. Veale & F. A. Cardoso (Eds.), *Computational creativity: The philosophy and engineering of autonomously creative systems* (pp. 193–207). Springer.

Hao, Y., & Veale, T. (2010). An Ironic Fist in a Velvet Glove: Creative Mis-Representation in the Construction of Ironic Similes. *Minds and Machines*, *20*(4), 483–488.

Hearst, M. (1992). Automatic Acquisition of Hyponyms from Large Text Corpora. In *Proceedings of the 14th conference on computational linguistics - volume 2* (pp. 539–545). COLING '92. Stroudsburg, PA, USA: Association for Computational Linguistics.

Lakoff, G., Espenson, J., & Schwartz, A. (1991). *The master metaphor list (Tech. Rep.)* University of California at Berkeley.

Lakoff, G., & Johnson, M. (1980). *Metaphors We Live By*. Chicago: University of Chicago Press.

Lavrač, N., Juršič, M., Sluban, B., Perovšek, M., Urbančič, T., & Cestnik, B. (2019). Bisociative knowledge discovery for cross-domain literature mining. In T. Veale & F. A. Cardoso (Eds.), *Computational creativity: The philosophy and engineering of autonomously creative systems* (pp. 119–138). Springer.

Martins, P., Pereira, F. C., & Cardoso, F. A. (2019). The nuts and bolts of conceptual blending: Multi-domain concept creation with Divago. In T. Veale & F. A. Cardoso (Eds.), *Computational creativity: The philosophy and engineering of autonomously creative systems* (pp. 91–118). Springer.

Pasca, M., & Durme, B. V. (2007). What You Seek is What You Get: Extraction of Class Attributes from Query Logs. In *Proceedings of the 20th international joint conference on artificial intelligence* (pp. 2832–2837). IJCAI'07.

Pereira, F. C. (2007). *Creativity and artificial intelligence: a conceptual blending approach*. Berlin: Walter de Gruyter.

Shutova, E., Sun, L., & Korhonen, A. (2010). Metaphor Identification Using Verb and Noun Clustering. In *Proceedings of COLING 2010* (pp. 1002–1010). Beijing, China.

Turney, P. D., & Littman, M. L. (2005). Corpus-based learning of analogies and semantic relations. *Machine Learning*, *60*(1-3), 251–278.

Veale, T. (2006). Tracking the Lexical Zeitgeist with Wikipedia and WordNet. In *Proceedings of ECAI'2006, the 17th European conference on artificial intelligence*, Trento, Italy: IOS Press.

Veale, T. (2012). *Exploding the Creativity Myth: The Computational Foundations of Linguistic Creativity*. London, UK: Bloomsbury.

Veale, T. (2013). Less Rhyme, More Reason: Knowledge-based Poetry Generation with Feeling, Insight and Wit. In *Proceedings of ICCC-2013, the 4th international conference on computational creativity*, Sydney, Australia.

Veale, T. (2014). A service-oriented architecture for metaphor processing. In *Proceedings of the second workshop on metaphor in NLP* (pp. 52–60). Baltimore, Maryland.

Veale, T., & Hao, Y. (2007a). Comprehending and Generating Apt Metaphors: A Web-driven, Case-based Approach to Figurative Language. In *Proceedings of AAAI'2007, the 22nd conference of the association for the advancement of artificial intelligence*.

Veale, T., & Hao, Y. (2007b). Making Lexical Ontologies Functional and Context-Sensitive. In *Proceedings of ACL'2007, the 46th annual meeting of the association for computational linguistics: Human language technologies*.

Veale, T., & Hao, Y. (2008). A Fluid Knowledge Representation for Understanding and Generating Creative Metaphors. In *Proceedings of COLING 2008* (pp. 945–952). Manchester, UK.

Veale, T., & Keane, M. (1997). The Competence of Sub-Optimal Structure Mapping on "Hard" Analogies. In *Proceedings of IJCAI'97, the 15th international joint conference on artificial intelligence*, San Mateo, California: Morgan Kaufmann.

Veale, T., & Li, G. (2011). Creative introspection and knowledge acquisition: Learning about the world thru introspective questions and exploratory metaphors. In *Proceedings of AAAI'2011, the 25th conference of the association for the advancement of artificial intelligence*, AAAI press.

Veale, T., & Li, G. (2013). Creating Similarity: Lateral Thinking for Vertical Similarity Judgments. In *Proceedings of ACL 2013, the 51st annual meeting of the association for computational linguistics*.

Veale, T., & O'Donoghue, D. (2000). Computation and Blending. *Cognitive Linguistics*, *11*, 253–282.

Veale, T., Shutova, E., & Klebanov, B. B. (2016). *Metaphor: A Computational Perspective*. USA: Morgan Claypool: Synthesis Lectures on Human Language Technologies.

Winston, P. (1980). Learning and reasoning by analogy. *Communications of the ACM*, *23*(12), 689–703.

Chapter 5
The Nuts and Bolts of Conceptual Blending: Multidomain Concept Creation with Divago

Pedro Martins, Francisco C. Pereira and F. Amílcar Cardoso

Abstract We revisit Divago, one of the first computational systems based on conceptual blending theory, along with an integrated and extended description of the main aspects that characterise it. Our tour around this framework includes revisiting past publications that report work related to the Divago architecture since the initial sketch presented in the late 1990s, up to publications reporting experiments in concept creation, including a detailed description of a more mature version introduced in 2005. Additionally, we report ongoing research work aimed at adding new relevant features to the system.

5.1 Introduction

The study of the phenomenon of creativity has led to the development of several models and psycho-cognitive theories, which, despite their differences, share the common principle that there is a strong relationship between creative thought and the ability to establish relations between seemingly unrelated domains of knowledge. For example, Guilford (1950) proposed *divergent thinking* as a key component of creativity. Guilford characterises divergent thinking as a process with the ability to generate multiple novel ideas as well as different solutions to a problem in a short period of time. Another example is Arthur Koestler's model of creativity, which relies on the *bisociation* mechanism (Koestler, 1964). Koestler coined the term 'bisociation' to describe the process by which two semantically distant matrices of thought are

Pedro Martins
CISUC, DEI, University of Coimbra, Portugal. E-mail: pjmm@dei.uc.pt

Francisco C. Pereira
Technical University of Denmark (DTU), Bygningstorvet 115, 2800 Kongens-Lyngby, Denmark. E-mail: camara@dtu.dk

F. Amílcar Cardoso
CISUC, DEI, University of Coimbra, Portugal. E-mail: amilcar@dei.uc.pt

T. Veale, F. A. Cardoso (eds.), *Computational Creativity*, Computational Synthesis and Creative Systems, https://doi.org/10.1007/978-3-319-43610-4_5

juxtaposed, enabling relations between seemingly unrelated pieces of information. According to Koestler, this process is the source of human creativity, whether it is in art, science or humour.

Designing computational models of mechanisms such as divergent thinking or bisociation is a considerably challenging task, owing to the difficulty in modelling complex elements like intuition or expectation. In any case, such mechanisms can be modelled to some extent in a multidomain context, where a computational reasoning system has access to a heterogeneous knowledge base where pieces of information related to different specific domains, with minor or no explicit relations between them, coexist. However, translating meaning becomes worthless if there is not a proper *integration* of the transferred knowledge into the novel context. The process should result in the generation of knowledge structures that can be considered as a whole, with an emergent structure, rather than a simple concatenation of its parts.

With regard to integration, it is crucial to have mechanisms with the ability to generate new structures that may extend the already existing ones. There are two main theories revolving around the process of integration: *conceptual combination* and *conceptual blending* (CB), which is also known as *conceptual integration*. The former focuses on the study of *noun–noun combinations* (such as 'elephant fish', 'street gun' or 'chair ladder') in the context of linguistics (Keane & Costello, 2001), whereas the latter discloses a process by which concepts are blended (or integrated), giving rise to an emergent structure (Fauconnier & Turner, 2002).

Although the CB framework cannot be viewed as a model of creativity per se, as the theory is sometimes vague and less conducive to formalisation when dealing with crucial aspects of creative thought, CB theory provides not only a comprehensive description of the so-called integration process but also a set of consistent principles as well as a terminology that can be used in creativity modelling. As a result, the CB framework has been the basis for several artificial creative systems.

In this chapter, we revisit Divago (Pereira, 2005), one of the first computational systems based on the CB framework, and present an integrated and extended overview of the main aspects that characterise it. Our tour around this creative system includes revisiting past publications that report work related to the Divago architecture since the initial sketch presented in the late 1990s (Pereira & Cardoso, 1999), up to publications reporting experiments in concept creation e.g., Pereira and Cardoso (2006), including a thorough description of a more mature version introduced in 2005 (Pereira, 2005, 2007). Additionally, we report ongoing research work aimed at adding new relevant features to the system.

The remainder of this chapter is organised as follows. Section 5.2 provides an overview of CB theory. Section 5.3 presents a description of several computational approaches to CB theory. Section 5.4 revisits the Divago framework and Section 5.4.1 describes an experiment developed with it. Finally, Section 5.5 draws final conclusions.

5.2 The CB Framework: An Overview

A key element in CB theory is the *mental space*, a term coined by Fauconnier to describe a partial and temporary knowledge structure created for the purpose of local understanding (Fauconnier, 1994). Mental spaces differ from *frames*, which are more stable knowledge structures. More precisely, frames encapsulate prototypes of entities, actions and reasoning, whereas mental spaces are particular short-term constructs whose structure is provided by the frames. For example, the mental space 'Mary's wedding' is organised by the 'marriage' frame.

In the CB framework, there is a network comprising at least four connected mental spaces (Fig. 5.1). Two or more of them correspond to the *input spaces*, which are the initial mental spaces. Then, a partial matching between the input spaces is constructed by connecting counterparts in the input spaces. This association is reflected in another mental space, the *generic space*, which contains elements common to the different input spaces, capturing the conceptual structure that is shared by them. The outcome of the blending process is the *blend space*, a mental space that maintains partial structures from the input spaces combined with an emergent structure of its own.

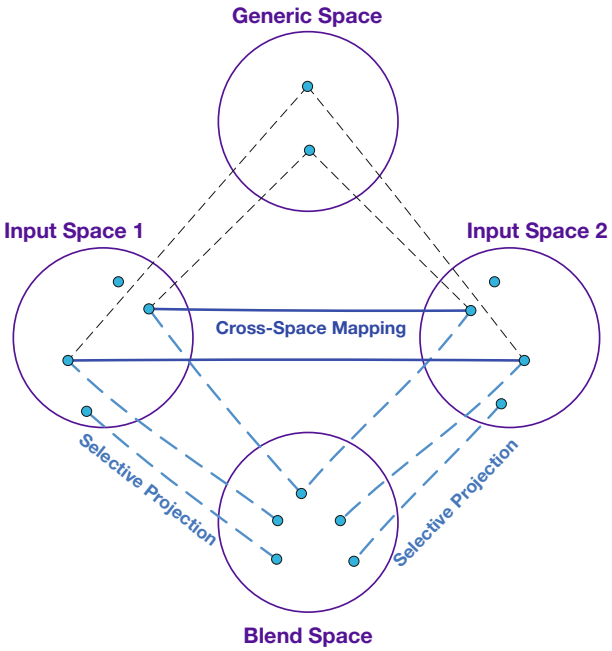


Fig. 5.1 The original four-space conceptual blending network (Fauconnier & Turner, 2002).

We give an example of conceptual blending in Fig. 5.2, where the concept 'computer virus' results from the blending of two mental spaces: 'computer' and 'virus'.

For the sake of simplicity, the generic space has been omitted. We can see the existence of an initial mapping (represented by dense dashed lines): 'computer' ↔ 'host' and 'program' ↔ 'virus'. This corresponds to making the analogy of a computer with a host (and a virus with a computer program). From these correspondences, selective projections from each input space are made into the new blended space: for example, the elements 'computer', 'instruction', 'program' and 'binary' are projected from the space 'computer', while the elements 'virus', 'resources', 'replicates', 'capacity' and 'unwanted' are projected from the space 'virus'; relations between these elements are also selectively projected from both spaces. The result is a blended space for a new domain that describes what we know as a 'computer virus'. The blended space borrows pieces from the two input spaces and, at the same time, has its own emerging structure. Note that a more accurate example of the 'computer virus' would have included other content from background knowledge.

5.2.1 Integration Process

According to CB theory, the integration of input elements in the blend space is performed through three operations: *composition*, *completion* and *elaboration*. Composition occurs when the elements from the input spaces are projected into the blend and new relations become available in the blended space. This implies projecting into the blend not only the matched elements but also other surrounding elements, as illustrated by the example just provided. Completion is an inferential operation that occurs when existing knowledge in long-term memory, i.e. knowledge from related background frames, is used to generate consistent and meaningful structures in the blend. Elaboration is an operation closely related to completion; it involves cognitive work to perform a simulation of the blended space. Elaboration is also known as 'running the blend'. There is not a pre-established order for these operations, and several iterations may occur.

5.2.2 Optimality Principles

While the possibilities for blending are apparently unlimited, not all blends are 'good'. Integration is guided by *optimality principles* (Fauconnier & Turner, 1998), which are responsible for generating the so-called 'good blends', i.e. consistent blends which are more easily interpreted. Fauconnier and Turner provided a list of these principles. For example, the *integration principle* states that the blend must be recognised as a unit. Another example is the *unpacking principle*, which requires that the blend alone must enable the 'blend reader'/observer to unpack the blend to reconstruct the inputs, the cross-space mapping, the generic space and the network of connections between all these spaces. There is also the principle of *relevance*, which requires the existence of a reason for the blend to occur. Other principles such as *topology*, *web*

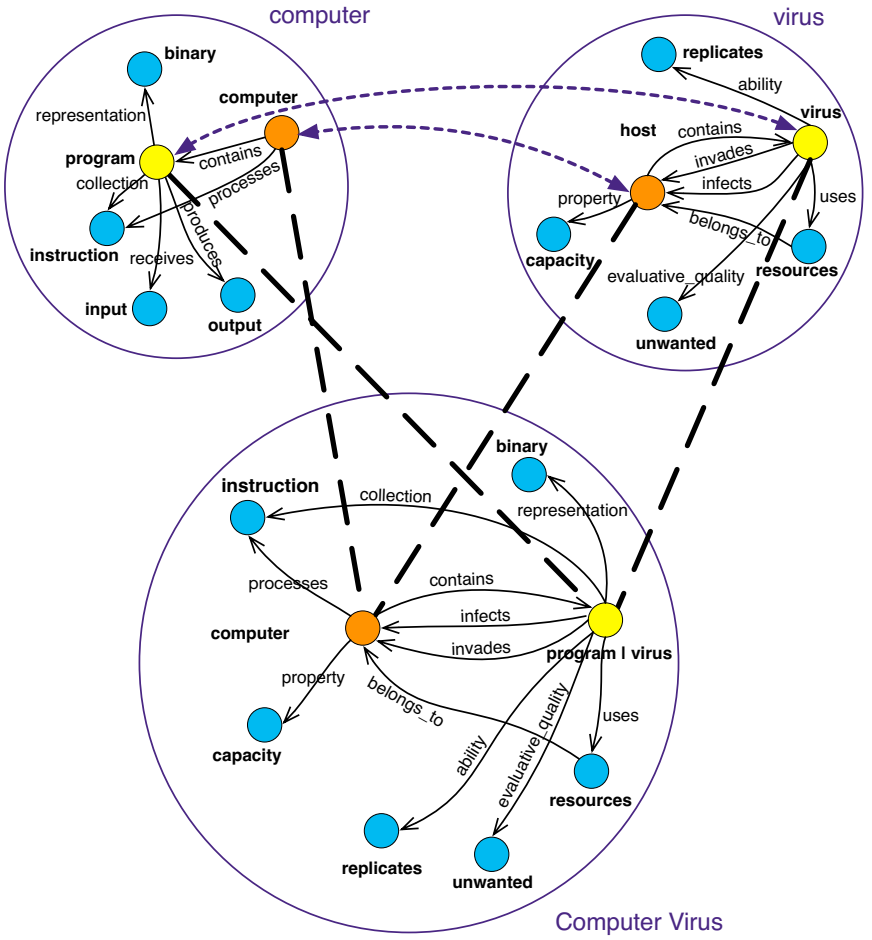


Fig. 5.2 'Computer virus' blend.

and *pattern completion* are responsible for managing the relationship between the input spaces and the blend.

Grady, Oakley, and Coulson (1999) suggested an additional principle of *metonymic tightening*, according to which "relationships between elements from the same input space should become as close as possible within the blend".

5.3 Computational Approaches to Conceptual Blending

Several authors have proposed computational models for CB following quite different approaches, including neuro-computational modelling. In this section, we present a

brief description of relevant work in the formalisation and computational modelling of CB theory. We will not refer to the Divago framework in this overview, as it will be described in more detail in the upcoming section.

Goguen (1999) presented one of the first attempts to formalise CB theory. Although it is more of a formalisation than a strictly computational approach, Goguen's seminal work has been a key influence on the design of computational models of the CB theory. Goguen proposed to describe the blending mechanism using the *algebraic semiotics* formalism. Algebraic semiotics is a formal theory combining algebraic abstract data type theory and ideas from the social sciences aimed at characterising *(systems of) signs* and mappings between them. In Goguen's formalisation, input spaces are treated as sign systems, whereas blends are *semiotic morphisms* between those systems. Goguen argues that the category theory operations of $\frac{3}{2}$ *co-limits* and $\frac{3}{2}$ *push-outs* provide the 'best possible blends' in a sense that includes ordering semiotic morphisms by quality.

According to Goguen, not all aspects of the CB mechanism can be formalised, such as some optimality principles, as they require human judgement and cannot be implemented in a straightforward manner.

The framework proposed by Veale and O'Donoghue (2000) is also one of the earliest computational approaches to CB. It relies on a model of analogy and metaphor, termed Sapper (Veale, 1995; Veale & Keane, 1997), to dynamically blend two domains. Sapper performs a structural integration by computing a mapping between two different domains (represented as graphs of concepts). The framework has the ability to decide which knowledge becomes relevant to a blend, and this is probably the most important aspect of the approach. However, it cannot be seen as a very detailed computational model of CB, since it does not cover some of the CB mechanisms and it is not clear whether the blends are a new space with an emerging structure or simply a mapping between the input spaces (Pereira, 2007).

Thagard and Stewart (2010) introduced a neuro-computational approach based on a mechanism that combines neural activity patterns by a process of *convolution* (a mathematical operation that interweaves structures). The main idea is to combine neural patterns into ones that are probably useful and novel. The motivation behind this work was to model the so-called *AHA! moment*, which occurs when humans discover surprising relations between seemingly unrelated pieces of information.

Concepts are represented as activity patterns of vectors of neurons, which are convoluted in order to combine patterns. Although Thagard and Stewart do not explicitly claim that this approach models the CB mechanism, they highlight the similarities between the proposed account of creativity and the blending mechanism. A key feature of this model is the ability to combine several multimodal representations, encompassing information that can be sensory, kinesthetic and verbal, as well as emotional. As for the latter, it is worth mentioning that emotional reactions play a key role in creative thought, in particular, the reaction of pleasure/approval that is associated with the generation of novel ideas.

Most of the current computational approaches to CB have a serious bottleneck: the need for a hand-crafted knowledge base. Veale (2012) proposed a framework to build *conceptual mash-ups*, which aim to be a robust, although limited, form of

blending that can produced by creatively reusing and extending existing common-sense knowledge about a certain topic, mined from available large corpora and the web, thus avoiding hand-crafted representations. The framework, which is described in more detail in another chapter of this book (Veale, 2019), is built on the idea that a creative system needs a critical mass of common-sense knowledge before applying it to a blending process, and must be able to collect it by itself.

Veale shows, in particular, that it is possible to mine knowledge snippets about concepts by performing clever queries in the Google API (what he calls 'milking' knowledge). For example, incomplete queries like 'why do poets *' and the retrieval of Google's completions allow the collection of common-sense knowledge about poets, like 'poets repeat words', 'poets use metaphors' and 'poets die young'. With these snippets, it is possible to apply introspection to explore possible blends by looking for evidence for possible common characteristics. For instance, to explore the blend '`poet|philosopher`' the program may check the number of results that the Google API counts for queries like 'philosophers repeat words' (0 results), 'philosophers use metaphors' (156 results) and 'philosophers die young' (6 results), which points to 'use metaphors' as a candidate characteristic for the blend.

Guhe et al. (2011) presented an account of blending based on *heuristic-driven theory projection* (HDTP) (Gust, Kühnberger, & Schmid, 2006; Schwering, Krumnack, Kuehnberger, & Gust, 2009), which was originally used as a framework for analogy making. HDTP represents knowledge about the domains as *first-order logic theories*, whose analogical mapping is established via *anti-unification*, i.e. an analogical relation is built by associating terms with a common generalisation.

Martinez, Besold, and Abdel-Fattah (2011) and Martinez et al. (2012) developed a blending algorithm by exploring a description of the input spaces based on the HDTP framework. They illustrated the applicability of the proposed model in considerably different scenarios such as mathematical domain formation, classical rationality puzzles and noun–noun combinations. It should be emphasised that although there is a detailed and precise description of the system, there is no clear evidence of an implementation.

Li, Zook, Davis, and Riedl (2012) addressed efficiency issues in blend generation and suggested a *goal-driven* and *context-driven* computational approach to CB. Their inspiration comes mainly from the work of Brandt and Brandt (2005), who studied the construction of *semiotic expressions*, a particular type of blend that is used in language to highlight certain aspects of one of the input spaces. For example, the expression 'this surgeon is a butcher' is a semantic one. Brandt and Brandt proposed *communication contexts* (e.g. interpersonal or small group communication) and goals as the driving force behind the processes of selection, projection, and elaboration. Li et al. authors extended this strategy to the generation of *standalone concepts*, a type of blend in which the blend is not meant to convey information about the input spaces. These authors also used the terminology proposed by Johnson-Laird (2002) to distinguish computational approaches where all possible blends have to be generated and individually tested (*neo-Darwinian algorithms*) from more efficient algorithms that generate only valuable blends by applying quality constraints on the search space (*neo-Lamarckian algorithms*). Li et al. presented two case studies of

systems that implement goal-driven and context-driven blending. The first case study focused on the problems of selective projection and elaboration and consisted of a system that constructs fictional gadgets in computer-generated stories. The second case study was mainly focused on the input selection: it was a system aimed at building objects used in a pretend play that result from the combination of features of a desired fantasy-world object with a real-world object.

The COINVENT project[1] (2013–2016) aimed to develop a computationally feasible, cognitively inspired formal model of conceptual blending (Schorlemmer et al., 2014). It revisited Goguen's ideas (Goguen, 1999, 2005) and intended to put conceptual blending on a firm mathematical ground, using *category theory*. In the project's framework, morphisms represent how the structures of input spaces are related, and conceptual blends correspond to co-limits. One of the important advantages of this approach is the high degree of domain independence of the techniques used, which makes it applicable to very different settings. As for application domains, the COINVENT team was primarily focused on mathematics and music, namely mathematical reasoning (Bou et al., 2015) and music harmonisation (Zacharakis, Kalikatso-Papakostas, & Cambouropoulos, 2015).

The formal model developed in the COINVENT project is closely related to the idea of *amalgamation*, i.e. the combination of several solutions from multiple cases (Besold & Plaza, 2015). The solution space becomes a generalised space that contains as much information from the input solutions as possible. When input solutions cannot be combined, amalgamation generalises them by omitting some of their details (Confalonieri, Corneli, Pease, Plaza, & Schorlemmer, 2015). Since combining and generalising solutions is computationally expensive, Confalonieri et al. (2015) proposed a discursive approach to evaluating the quality of blends. The idea is to use *Lakatosian argumentative dialogue* (Lakatos, 1976) to iteratively construct valuable and novel blends as opposed to a strictly combinatorial approach, i.e. blends are evaluated under a process of argumentation in which the particulars of a given blend are identified and presented as issues of discussion. The Lakatosian argumentative dialogue is a model of argumentation, which is presented as a dialogue to describe the different ways in which mathematicians establish new theories.

Also within the scope of the COINVENT project, Kutz, Neuhaus, Mossakowski, and Codescu (2012) presented a basic formalisation of CB in which the *Distributed Ontology Language* (Lange, Kutz, Mossakowski, & Grüninger, 2012) was used to specify blending diagrams. This work involved the translation of specifications initially written in the OBJ language (Goguen & Malcom, 1996).

5.4 Divago

One of the first attempts to computationally model the CB framework was Divago, started in the late 1990s (Pereira & Cardoso, 1999), which evolved into an elaborate

[1] www.coinvent-project.eu

and extensive tool that is still an object of research. The system was named after the Portuguese expression 'Eu divago', which means 'I wander' or 'I digress'. The name emphasises the importance of divergence in creative thought, i.e. the ability to find unexpected solutions by avoiding common or biased reasoning. However, Divago is more than an implementation of divergent reasoning; it encapsulates the divergent and convergent parts of a creative process: the blended space is the result of integrating two domains. Furthermore, it is still the only CB-based system to date that uses an explicit formalisation of the optimality principles to guide blend generation.

The architecture of Divago is depicted in Fig. 5.3. The system is composed of several modules and works on a *(multidomain) knowledge base*. We will start our description by analysing how the knowledge base is structured. The operation of the remaining modules will be described afterwards.

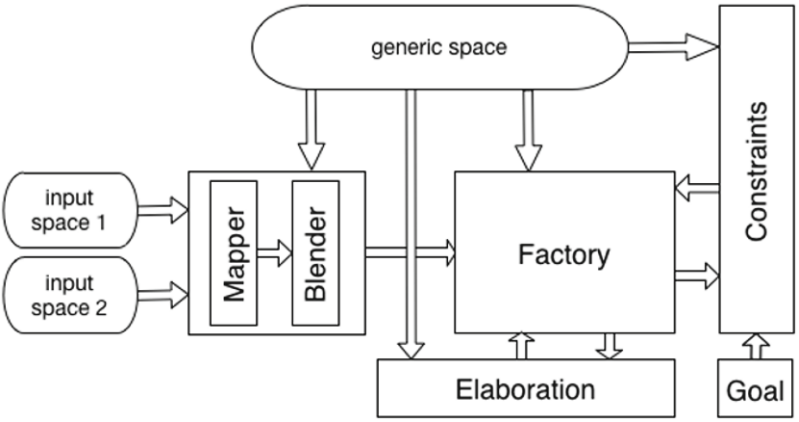


Fig. 5.3 Divago's architecture.

The knowledge base is a dynamic repository of information divided into *domains*. Each domain includes several types of knowledge: a *concept map*, a set of *instances*, a set of *rules*, a set of *integrity constraints* and a set of *frames* (Pereira & Cardoso, 2006).

A *concept map* is a semantic network that denotes the relationship between the concepts of a given domain, which correspond to the elements of a mental space in the CB framework. It corresponds to the factual part of the micro-theory of the domain. We often represent concept maps graphically, as graphs in which the relations are arcs and elements are nodes. Figure 5.4 shows excerpts of the concept maps for 'horse' and 'bird'. In these examples, one can observe typical relations such as *part–whole* (pw), *purpose*, *property* (ppty) or *quantity* (qty).

An *instance* is a specific example of the domain. For example, a particular description of a horse is an instance of the domain *horse*.

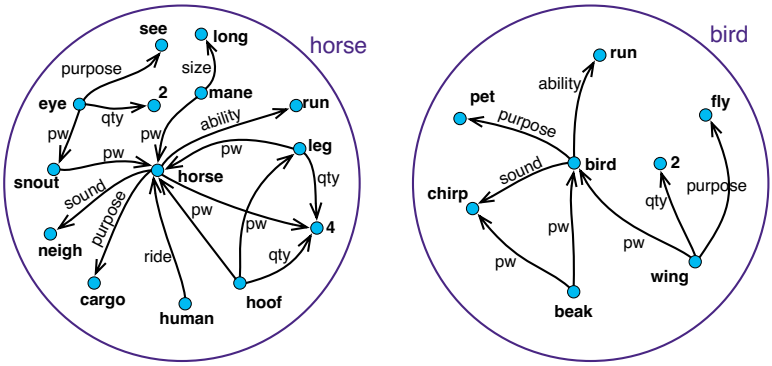


Fig. 5.4 Concept maps of horse and bird (excerpts).

Rules are used to represent inherent causality. For example, a rule for inferring that something is in the gaseous state (and not in the solid or liquid state) could be

$$state(X, gaseous) \wedge not\ state(X, solid) \wedge not\ state(X, liquid) \longleftarrow ebullition\_lvl(X, N) \wedge temperature(X, T) \wedge T > N$$

Integrity constraints are particular cases of rules that serve to assess the consistency of the concept by describing events or facts that cannot occur simultaneously. For example, an integrity constraint could express that *an object X cannot be solid and liquid at the same time*.

Frames are a type of knowledge that encompasses a set of conditions and guidelines that define properties of the blend to be generated. They have the role of describing abstract prototypes of entities, actions, reasoning, situations or idiosyncrasies. A frame consists of a set of conditions that the concept map must satisfy. When a concept c satisfies all the conditions of a frame f, we say that c *integrates* f. For example, the frame of 'transport means' corresponds to a set of elements and relations that, when connected together, represent something that has a container and a subpart (e.g. an engine) that serves for locomotion:

$$frame(transport\_means(X)) : carrier(X, people) \longleftarrow have(X, container) \wedge have(X, Y) \wedge purpose(Y, locomotion) \wedge drive(\_, X)$$

When a concept map *integrates* the 'transport means' frame, then we can say that either it is itself a 'transport means' or one of its constituents is a 'transport means'. For example, the concept map of a school bus would integrate this frame, while the concept map of a classroom would not.

Having described the knowledge base, let us now focus on how Divago's modules operate. The first step in the concept invention process is the selection of the input knowledge, which corresponds to choosing a pair of input spaces (domains) from

the knowledge base. In its original implementation, this selection was not made proactively by Divago and had to be performed by some external means (e.g. by a human user). Then, the *Mapper* module performs the selection of elements for projection. This selection is achieved by means of a partial mapping between the input spaces using *structural alignment*. This operation looks for the largest *isomorphic* (structurally equivalent) pair of subgraphs contained in the input spaces. Here, structural equivalence means that the graphs have the same edges (relations) regardless of the nodes. Figure 5.5 illustrates a mapping between two elementary domains – *horse* and *bird* – via structural alignment.

For each mapping provided by the Mapper, the *Blender* performs a projection into the blend space. All the possible projections resulting from each mapping must be represented in the blend space at this stage, as we want to be as exhaustive as possible. Our algorithm first projects the nodes involved in the mapping (see Fig. 5.6). For example, the mapping 'wing' $\leftrightarrow$ 'hoof' has four alternative projections: each node is projected as a separate concept (nodes 'hoof' and 'wing' in the blend), no node is projected (a '*nil*' node in the blend space) or both nodes are combined into a node for a new concept, '$wing||hoof$'. Each of these four combinations may be a part of a possible blend. Then, non-mapped nodes are projected as a copy of themselves and as a *nil* node (meaning that the node may appear or not in each of the possible blends), as shown in Fig. 5.7). At last, the remaining elements in the input spaces (in this example, the relations) are also projected, as depicted in Fig. 5.8 (for simplicity, we omit the *nil* nodes in this figure). The whole set of projections summarise the set of all possible blends, which is called the *blendoid*. In a realistic example, the blendoid contains all possible relations, rules, frames, instances and integrity constraints that can be present in any blend of two given input spaces.

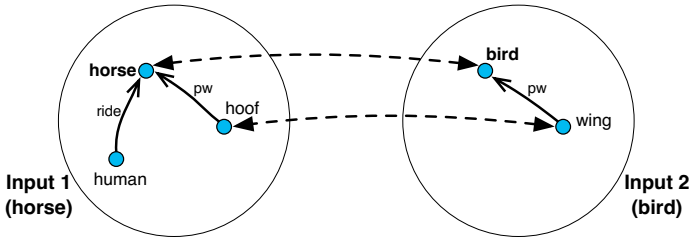


Fig. 5.5 Selecting elements for projection.

The *Factory* module is responsible for exploring the space of all possible blends produced by the Blender. The Factory interacts with both the *Elaboration* and the *Constraints* modules: it is based on a genetic algorithm (GA) that looks for the *elaborations* of blends that best fulfil the optimality principles proposed in the theory. The Constraints module implements the optimality principles into a set of constraints. In each iteration, the GA sends each blend to the Elaboration module, which is responsible for applying context-dependent knowledge and thus enriching the blend. Then it sends the result to the Constraints module, which applies the optimality

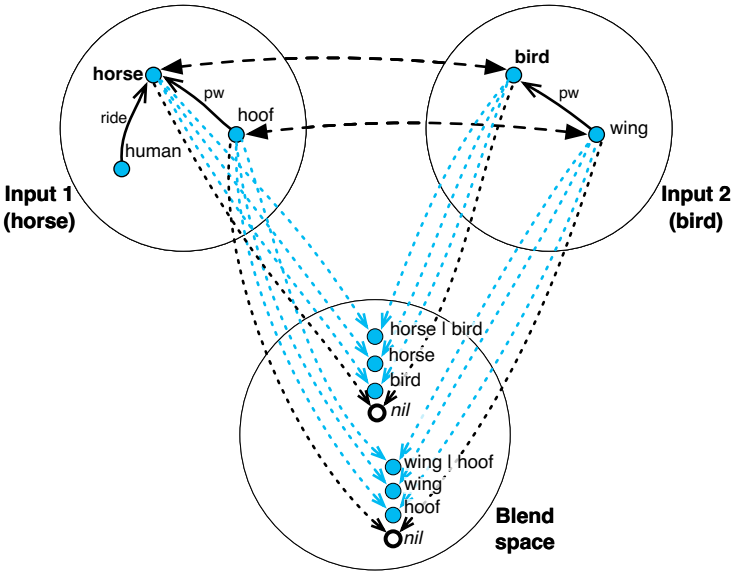


Fig. 5.6 Projecting *mapped nodes* into the blend.

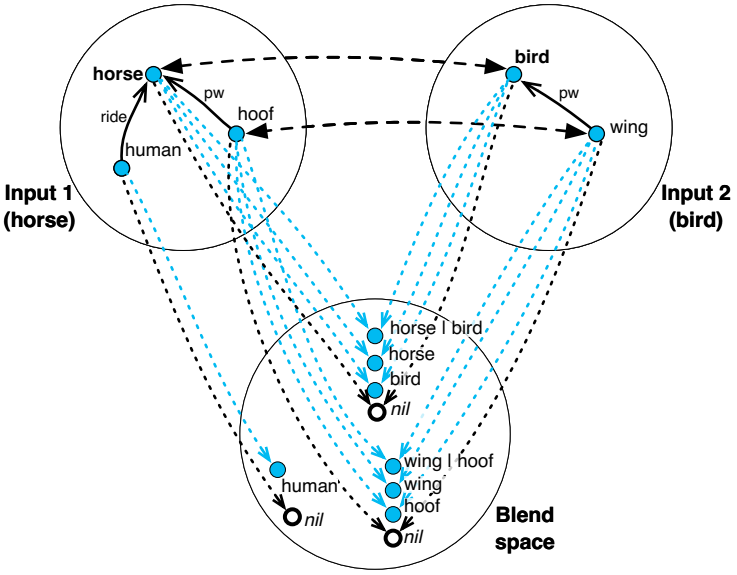


Fig. 5.7 Projecting *non-mapped nodes* into the blend.

constraints to the blend to assess it. This module provides, therefore, the fitness function for the evolutionary process. The optimality constraints can be seen as *competing pressures* on the evolutionary process.

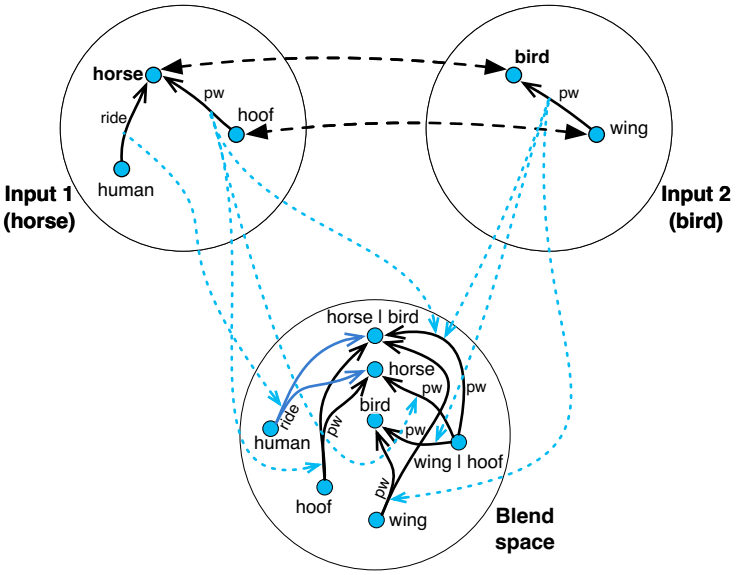


Fig. 5.8 Projecting *relations* elements into the blend.

When the GA finds an adequate solution or a pre-defined number of iterations is reached, the Factory stops the execution of the GA and returns the best blend.

A key feature of the Divago system is the explicit use of the optimality principles in the blending process. Note that several other computational models of CB do not explicitly implement this component. In the case of Divago, seven principles are modelled and measures for each one of them are provided (*Integration*, *Topology*, *Unpacking*, *Maximisation/Intensification of Vital Relations*, *Web*, *Relevance* and *Pattern Completion*):

- *Integration*. This principle states that the blend must constitute a tightly integrated scene that can be manipulated and perceived as a unit. In Divago, integration is driven by frames, which gather knowledge around abstractions, tightening the links between elements. They organise a concept into a more understandable whole. We define the *frame coverage* of a blend to be the set of relations from its concept map that belong to the set of conditions of one or more frames that are satisfied in the blend. The *integration value* increases with the frame coverage of the blend, and decreases with the number of frames involved in the coverage. The intuition behind this is that the unity around an integrating concept (the frame) reflects the unity of the blend; however, a blend that is covered by many frames should be less integrated than a blend with the same coverage but with fewer frames.
- *Topology*. This principle states that for any input space and any element in that space projected into the blend, it is optimal for the relations of the element in the blend to match the relations of its counterpart. In Divago, the value of Topology

corresponds to the ratio of topologically correct relations in the concept map of the blend, where a relation in a blend is said to be *topologically correct* when it occurs in one of the input spaces. This constraint brings inertia to the blending process, i.e. it drives against change in the concepts, as in order to maintain the same topological configuration as in the inputs, the blend needs to maintain exactly the same neighbourhood relationships between every element. Full application of this constraint would make the blend a projected copy of the inputs. This constraint is particularly useful when the problem at hand involves analogical construction, but is secondary if it concerns a free combination of concepts, where novel associations are more tolerable.

- *Unpacking*. This principle refers to the ability of the blend to enable the 'blend reader' to recognise and reconstruct the input spaces. In Divago, this is understood as the ability to reconstruct the cross-space mapping, the generic space and the network of connections between all these spaces. So, the estimated Unpacking value of a blend is computed as a measure of the presence of relations from the *defining frames* of the elements in the blend. Here, a defining frame corresponds to a frame comprising the immediate surrounding elements and relations.
- *Maximisation/Intensification of Vital Relations*. The CB theory proposes a set of vital relations that should govern the process of blend creation (Fauconnier & Turner, 2002). Divago also accepts the definition of other relations as being vital. For example, in inventing concepts for a game, one may decide that the vital relations are 'strength', 'defence', 'ability' and so on. The effect of this choice may be that, when one gives more emphasis (higher weight in the fitness function) to Maximisation of Vital Relations, the resulting blends will contain the maximum possible number of these relations. The impact of the vital relations on a blend is estimated by computing the ratio of vital relations in the blend with respect to the whole set of possible vital relations contained within the blendoid.
- *Web*. This principle states that when the blend is manipulated as a unit, it must maintain the web of appropriate connections to the input spaces easily and without additional surveillance or computation. This principle is co-related to Topology and Unpacking: the former provides a straightforward way to 'maintain the web of appropriate connections to the input spaces easily and without additional surveillance or computation', while the latter measures exactly the work needed to reconstruct the inputs from the blend. It is not, therefore, an independent principle. In Divago, Web is estimated by combining the estimates for Topology and Unpacking in a weighted sum.
- *Relevance*. All things being equal, if an element appears in the blend, there will be pressure to find significance for this element. A blend, or a part of it, may be more or less relevant depending on what it is for. Given a set of *goal frames*, which can be selected from the ones available in the knowledge base or specified externally, a blend has the maximum Relevance value if it is able to satisfy all of them. From the point of view of creativity, we propose the use of Relevance as a *usefulness* measure, an idea that has been applied in some experiments.

- *Pattern Completion*. Other things being equal, complete elements in the blend by using existing integrated patterns as additional inputs. As in Divago a pattern is described by a frame, pattern completion corresponds essentially to frame completion.

All the measures above are normalised to the interval [0,1].

5.4.1 The Horse-Bird Experiment

In the course of its life, Divago was subject to several different experiments, with complementary aims. The experiments involved knowledge from a diversity of domains, in a strategy aimed at minimising the risks of fine-tuning the system and biasing the results. For this chapter, we have selected an experiment aimed at better understanding the individual effects of each one of the optimality constraints on the results of the evolutionary process: the horse-bird experiment (Pereira & Cardoso, 2003). The results of the experiment also allowed a qualitative evaluation of the model and its tuning.

As we did not know in advance the distribution of the populations involved in the experiment, we followed the Central Limit Theorem and decided to rely on the condition that each sample must be large, so each of the optimality constraint weight configurations tested was subjected to 30 runs with the same starting conditions, each run being an entire evolutionary cycle, from the initial population to the population at which the algorithm stopped. After several preliminary tests of the tuning of the GA parameters, we decided on 100 individuals as the population size, with 5% of asexual reproduction (a copy of an individual to the following population), 80% of crossover (combination of pairs of individuals), 14% of mutation and 1% of random generation (to allow random jumps in the search space). We used three different stopping conditions: the appearance of an individual with the maximum value (1); achieving n populations ($n = 500$); and being stalled (no improvements in best value) for more than m populations ($m = 20$). We kept these GA configurations throughout the whole experiment described here.

Given the purpose of the experiment, the Elaboration module was not used, as we did not want it to mask any effect of the constraints on the result. Therefore, each blend was examined by the Constraints module without being subject to any transformation after the projections.

5.4.1.1 Evaluating the Optimality Constraints

The Constraints module computes the fitness of a blend through a weighted sum of the outputs of each optimality constraint applied to the blend. To analyse the role of each individual constraint, we isolated each one by attributing zero weight to the remaining constraints.

The input domains we applied were the domains of *horse* and *bird* (Tables 5.1 and 5.2), meaning that the expected results ranged from an unchanged copy of one (or both) of the concepts to a horse-bird (or bird-horse), which would be a combination of selected features from the input domains.

Table 5.1 The concept map of *horse*

isa(horse, equinae)	pw(leg, horse)	purpose(horse, food)
isa(equinae, mammal)	purpose(leg, stand)	sound(horse, neigh)
existence(horse, farm)	pw(hoof, leg)	purpose(mouth, eat)
existence(horse, wilderness)	purpose(horse, traction)	purpose(ear, hear)
pw(snout, horse)	eat(horse, grass)	color(mane, dark)
pw(mane, horse)	ability(horse, run)	size(mane, long)
pw(tail, horse)	carrier(horse, human)	material(mane, hair)
quantity(hoof, 4)	quantity(leg, 4)	purpose(horse, cargo)
pw(eye, snout)	quantity(eye, 2)	taxonomic(horse, ruminant)
pw(ear, snout)	quantity(ear, 2)	ride(human, horse)
pw(mouth,snout)	purpose(eye, see)	motion_process(horse, walk)
isa(farm, human_setting)		

Table 5.2 The concept map of *bird*

isa(bird, aves)	existence(bird, house)	isa(aves, oviparous)
lay(oviparous, egg)	existence(bird, wilderness)	purpose(bird, pet)
purpose(bird, food)	purpose(eye, see)	smaller_than(bird, human)
pw(lung, bird)	motion_process(bird, fly)	purpose(beak, chirp)
purpose(lung, breathe)	quantity(eye, 2)	quantity(wing, 2)
isa(owl, bird)	isa(paradise_bird, bird)	quantity(claw, 2)
ability(bird, fly)	pw(wing, bird)	conditional(wing, fly)
pw(feathers, bird)	pw(beak, bird)	purpose(wing, fly)
purpose(beak, eat)	purpose(claw, catch)	sound(bird, chirp)
isa(parrot, bird)	ability(parrot, speak)	pw(straw, nest)
pw(eye, bird)	pw(leg, bird)	purpose(leg, stand)
pw(claw, leg)	role_playing(bird, freedom)	quantity(leg, 2)
isa(nest, container)	isa(house, human_setting)	

We applied the three mappings presented in Table 5.3. For each mapping, we tested the six optimality pressures, each of these comprising 30 runs.

We now present a detailed analysis of the individual effect of each of the measures (all values normalised to the interval [0,1]):

- In *Integration*, frames behave as *attractor* points in the search space. The complexity of the search space grows with the mapping size (the number of cross-space associations found by the mapping algorithm). In fact, when we have a mapping of size 5, it returns six different blends, with the best choice being retrieved 43% of the time, while with a mapping size of 21, it finds eight different solutions, with the best choice being retrieved only 6% of the time. A good compensation for this apparent loss of control is that the returned values are

Table 5.3 The three mappings

		vegetable_food ↔ vegetable
		food ↔ food
ear ↔ wing		horse ↔ bird
snout ↔ bird		equidean ↔ aves
eye ↔ lung		animal ↔ animal
mouth ↔ feathers		human_setting ↔ house
2 ↔ 2		wilderness ↔ wilderness
hear ↔ fly		ruminant ↔ oviparous
		run ↔ fly
1		cargo ↔ pet
		neigh ↔ chirp
		snout ↔ lung
		mane ↔ feathers
		tail ↔ beak
mouth ↔ beak		leg ↔ eye
snout ↔ bird		hoof ↔ wing
eye ↔ lung		4 ↔ 2
ear ↔ feathers		eye ↔ leg
eat ↔ eat		ear ↔ claw
		hear ↔ catch
2		grass ↔ grass
	3	

clearly higher (0.68, for the best) than for the small mappings (0.22), suggesting that, with larger mappings, the probability of finding a better solution is higher than for smaller ones.

- *Pattern Completion* drives the blend to partially complete (i.e., satisfy some but not all conditions) the highest possible number of frames, leading, in each case, to several sets of relations that fit into those frames without satisfying them. This means that, in isolation, Pattern Completion leads only to disperse, non-integrated results and so is not very useful. Interestingly, it can be useful when combined with Integration because it brings gradually to the blend the concepts and relations that are needed to complete the frames and so speeds up the process of finding frames with a high Integration value. Its search landscape seems to be very rich in local maxima. The most constant results came from mapping 2 (in Table 5.3), with the best results obtained 13% of the time and the second best 20% of the time. An interesting point is that the resulting local maxima always fall within a very strict range of values (of maximum amplitude 0.11, in mapping 3).
- In all the experiments with *Topology*, the final results were valued at 100%, meaning that this constraint is easily fully accomplished, independently of the mapping. An interesting fact is that there is a multitude of solutions in the search landscape of Topology, demonstrated by the number of different final results in each mapping. Intuitively, and observing the short duration of each run, this means that, wherever the search starts, there is always a Topology-optimal point

in the neighbourhood. From observation of the relations contained in the final results, we see that this constraint brings a tendency towards *disintegration*, i.e. small isolated graphs appear in the blend. Each isolated graph either is a copy of a (normally unmapped) subgraph of one input source or consists of complete structure matching (there are concepts from both domains, but only the relations that exist in both are present).

- The influence of *Maximisation of Vital Relations* on the results is straightforward, given that its highest value (1) reflects the presence, in the blend, of all the vital relations that exist in the inputs. As the evolution goes on in each run, the value grows until it reaches the maximum reasonably early on. For each set of 30 runs, it reached the value 1 a minimum of 93% of the time, and the remaining 7% of the runs achieved at least a value of 0.95. As in the case of Topology, the search space of Maximisation of Vital Relations is very simple, since there is a global maximum in the neighbourhood of (almost) every point.
- The results for the *Unpacking* measure show that it has a deleterious side effect on the size of the blend, as it drives it to very small sets (between 0 and 5) of relations. The interpretation here is straightforward: the ratio of *unpackable* concepts is highly penalised in bigger sets because of the projected relations that come as a side effect of the projection of concepts (unpackable or not). These relations *confuse* the unpacking algorithm so that it leads the evolution to gradually select the smaller results. The maximum points also correspond to the value 1, but it seems from the experiments that there is a very limited set of such individuals, achieved in the majority (at least 93% for each mapping) of the experiments.
- The first part of the test for *Relevance* focused on making a single-relation query. In this case, we asked for 'something that flies' (ability(_, fly)). The results were straightforward in any mapping, accomplishing the maximum value (1) in 100% of the runs, although the resulting concept maps did not necessarily reveal any overall constant structure or unity, giving an idea of randomness in the choice of relations other than ability(_, fly). In other words, the evolution took only two steps: one where no individual had a relation 'ability(_, fly)', and therefore had a value 0; and on where a relation 'ability(_,fly)' was found, yielding a value of 1, independently of the rest of the concept map. The second part of the test for Relevance, by adding a frame (*ability_explanation*) to the query, revealed similar conclusions. There was not sufficient knowledge in any of the input domains to satisfy this new frame completely, so the algorithm searched for the maximum satisfaction and reached it 100% of time in every mapping. Thus, the *landscape* seems to have one single global maximum and no local maxima, reflecting the integration of the two parts of the query. If there were separate frames, the existence of local maxima would be expected. Intuitively, the *search landscapes* of Integration and Relevance seem to be similar.

5.4.1.2 Qualitative Evaluation

In this stage of the experiment, we tried to understand the behaviour of the system by generating and observing different blends produced when we gave specific goals to Divago. The first goal was to generate a *well-known* blend of a horse and a bird: the *Pegasus*. Then, we allowed more variations of this creature, by changing the mapping or the weights of the optimality pressures.

5.4.2 The Pegasus

For our purposes, we define a Pegasus as being a 'flying horse with wings', so leaving out other features it may have (such as being white). Formally, the Pegasus we want to generate has the same concept map as the horse domain augmented with two wings and the ability to fly (so, it should also have the relations $ability(horse, fly)$, $motion\_process(horse, fly)$, $pw(wing, horse)$ and $quantity(wing, 2)$).

For validation purposes, we started by submitting a query with all the relations of the Pegasus, to check if they could be found in the search space, and, obviously, the results reveal that only the mapping 3 (see Table 5.3) respects such constraints. This led us to use this mapping exclusively throughout the rest of the Pegasus test. Knowing that the solution existed in the search space, our goal was to find the minimal necessary requirements (the weights, the frames and the query) in order to retrieve it. From a first set of runs, in which the system considered a big set of different frames and no query, we quickly understood that it was not simple (or even feasible) to build the Pegasus solely by handling the weights. This happens because the optimality pressures provide control regarding structural evaluation and general consistency, but only by pure chance we can find the exact weights to match the relations of the Pegasus, a very specific blend that fails to follow only a few of the constraints, but a particular combination of them. This drives us to the need for *queries*.

A query may range from specific conditions that we require the blend to respect (e.g. the set of conditions for flying, enumerated above) to highly abstract frames that reflect our preferences in the blend construction (e.g. the frame *aprojection*: concepts from input concept map 1 should all be projected). Intuitively, the best options seem to comprise a combination of the different levels of abstraction.

Since a query is considered only in the Relevance measure, its weight must be large if we intend to give it priority. In fact, using only Relevance is sufficient to find the solution if the query is specific enough, as we could test by using a query with *aprojection* and the flying conditions. From the point of view of creativity, however, we do not expect to have very specific queries and we are more interested in less constrained search directives. In Table 5.4, we show the parameters we used. The weights we present correspond to Integrity (I), Pattern Completion (PC), Topology (T), Maximisation of Vital Relations (MVR), Unpacking (U) and Relevance (R). The

'fly conds.' are the relations the blend must have in order to be a flying creature, and aframe, aprojection and new_ability are frames (Pereira, 2005).

Table 5.4 The 10 different configurations used

Experiment No.	Weights						Query
	I	PC	T	MVR	U	R	
1	0	0	0	0	0	100	fly conds. + aprojection
2	0	0	0	0	0	100	fly conds. + aframe
3	0	0	0	0	0	100	fly conds. + aprojection + aframe
4	50	0	0	0	0	50	fly conds. + aprojection + aframe
5	33.3	33.3	0	0	0	33.3	fly conds. + aprojection + aframe
6	33.3	0	33.3	0	0	33.3	fly conds. + aprojection + aframe
7	25	0	25	25	0	25	fly conds. + aprojection + aframe
8	20	0	20	20	20	20	fly conds. + aprojection + aframe
9	34	0	16	10	4	36	fly conds. + aprojection + aframe
10	34	0	16	10	4	36	new_ability + aframe + aprojection

The first eight configurations were dedicated to understanding the effect of gradually adding optimality pressures to the fitness function. In the first three, where only Relevance was used, we verified that, although it was *easy* to have all the concepts and relations we expect for a Pegasus, often this was complemented by an apparently random selection of other relations. This results from having no weight on Integration, which we added in configuration 4, yielding the most strict Pegasus, the projection of the entire horse domain, and the selective projection of wings and the ability to fly from the bird domain, in more than 90% of the runs. In configuration 5, the influence of Pattern Completion led the results to minimum incompleteness (e.g. a Pegasus with everything except a mane, wings or any other item), which revealed that, by itself, Pattern Completion is not a significant or even positive contribution to the present goal, a reason for dropping its participation in the subsequent configurations.

Adding Topology (configuration 6) brought about essentially two different kinds of results. In 60% of the runs, it returned the 'correct' Pegasus with extra features like having feathers or a beak (which was not constrained in the query), either of them apparently selected at random. These were also given higher scores in the experiment. In another 37% of the runs, the results were either 'simple' horses or a compromise between a bird and a horse (e.g. two legs, a beak, two wings, ruminant, a mane or hooves). A possible interpretation is that, on one side, the frames *aprojection* and *aframe* already imply strong topological maintenance, and Topology itself brings knowledge that, although not considered in the frames, strengthens this value. Yet, this does not avoid the existence of local maxima that represent stable results, in terms of the weights considered. The following configuration, with the inclusion of Maximisation of Vital Relations, confirmed these conclusions, but with more control over the kind of extra relations transferred to the blend.

The eighth configuration brought about a more stable set of results. Adding Unpacking to the other pressures ensured the prominence of the 'basic' Pegasus, but, as happened with the results for the sixth configuration, it was augmented with

features projected from the bird domain. This time, some of these new features came isolated to the blend, i.e. not connected to the rest of the blend (e.g. there were two claws that served to catch, but they did not form part of anything).

An immediate conclusion we drew from these experiments was that each pressure should have a different weight, corresponding to the degree of influence it should have on the result. In our case, we were seeking a specific object (the Pegasus), and we knew what it was like, what it should not have and some features not covered by the query conditions that we would like it to have. This led us to a series of tests for obtaining a satisfiable set of weights, used in the configurations 9 to 12. Given the huge dimension of the problem of finding these weights, they were obtained from a generate-and-test process, driven by our intuition, so there is no detailed explanation for the exact choice of these values and not others.

Configuration 9 revealed, possibly, the 'best' Pegasus we could expect. As we can see in the two results presented in Tables 5.5 and 5.6, it has all the horse features, the specified 'flying' requirements and some added knowledge that we consider valid, like having two wings, lungs or claws. It is clear that these results were subjectively driven by us through the choice of the concepts and frame design, but the argument we are trying to make is that it produces a new concept that not only respects the query but also brings with it new knowledge that was selectively projected.

Table 5.5 Example 1 (from configuration 9)

quantity(wing, 2)	conditional(wing, fly)	motion_process(horse, fly)
ability(horse, fly)	purpose(wing, fly)	pw(wing, horse)
isa(horse, equinae)	pw(leg, horse)	purpose(horse, food)
isa(equinae, mammal)	purpose(leg, stand)	sound(horse, neigh)
existence(horse, farm)	pw(hoof, leg)	purpose(mouth, eat)
existence(horse, wilderness)	purpose(horse, traction)	purpose(ear, hear)
pw(snout, horse)	eat(horse, grass)	color(mane, dark)
pw(mane, horse)	ability(horse, run)	size(mane, long)
pw(tail, horse)	carrier(horse, human)	material(mane, hair)
quantity(hoof, 4)	quantity(leg, 4)	purpose(horse, cargo)
pw(eye, snout)	quantity(eye, 2)	taxonomic(horse, ruminant)
pw(ear, snout)	quantity(ear, 2)	ride(human, horse)
pw(mouth,snout)	purpose(eye, see)	motion_process(horse, walk)

In the final configuration (10), we decided to give a more vague query, asking only for a *new_ability* in the blend, as well as the generic frames *aprojection* and *aframe*. As a result, we found the exact Pegasus 23% of the time. This gives the first evidence that the system can be used for generating concepts without a very constraining and specific query and led us to the following experiments, in which we tried to assess its generative possibilities.

Table 5.6 Example 2 (from configuration 9)

purpose(claw, catch)	pw(claw, leg)	purpose(lung, breathe)
pw(lung, horse)	conditional(wing, fly)	motion_process(horse, fly)
ability(horse, fly)	purpose(wing, fly)	pw(wing, horse)
isa(horse, equinae)	pw(leg, horse)	purpose(horse, food)
isa(equinae, mammal)	purpose(leg, stand)	sound(horse, neigh)
existence(horse, farm)	pw(hoof, leg)	purpose(mouth, eat)
existence(horse, wilderness)	purpose(horse, traction)	purpose(ear, hear)
pw(snout, horse)	eat(horse, grass)	color(mane, dark)
pw(mane, horse)	ability(horse, run)	size(mane, long)
pw(tail, horse)	carrier(horse, human)	material(mane, hair)
quantity(hoof, 4)	quantity(leg, 4)	purpose(horse, cargo)
pw(eye, snout)	quantity(eye, 2)	taxonomic(horse, ruminant)
pw(ear, snout)	quantity(ear, 2)	ride(human, horse)
pw(mouth, snout)	purpose(eye, see)	motion_process(horse, walk)

5.4.3 Other Creatures

In order to explore the potential of the system, we made additional tests, without imposing specific goals beforehand. We did not make significant variations in the weights of the previous tests. For two, we removed some weights from the configuration and reduced Integration in the latest ones. In Table 5.7, we show all the configurations (the omitted weights are 0, as in the other experiments). We made variations in the query and checked the results, trying not to bias for particular outcomes. Therefore, these tests were aimed at giving an informal insight into the generative potential of the system.

We found several 'creatures' that we would like to describe. We call the first (configuration 11) a 'dumborse', a flying horse that uses its ears as wings (like Dumbo, the flying elephant). It is possible to find this 'creature' in mapping 1 (*ears* are mapped onto *wings*). It is exactly a horse, but it has wings instead of ears, which serve both to fly and to hear. With Dumbo in mind, we tried to go further to a horse with ears that serve to fly and hear (instead of wings in place of ears), and this was achieved by allowing only weights on Integration and Relevance (configuration 12). A simple explanation is that, while it satisfies Relevance entirely and Integration almost totally, it has less Topology and less Unpacking (ears do not ever relate to flying in the bird domain).

Another creature to report is the 'flying snout' (which appeared in 23% of the runs of configuration 13, see Table 5.7), a snout that has all the features of a bird. This is a 'weak' blend in the sense that an isolated concept (the 'snout') is projected into the 'bird' structure without any surrounding support such as its shape or its purpose. The third creature is the 'transport bird', which has all the features of a bird, but also carries humans; it serves for cargo and traction. It appeared occasionally during the previous experiments, but was triggered now by the frame 'transport_means' in the query (in configuration 20), meaning that we indeed had it in mind. Yet, its appearance throughout the tests (only when dealing with mapping 3, though) led us to include it in this section. The fourth creature is an oviparous horse, with two legs

(instead of four), two wings, and claws. It appeared in fewer than 10% of the results for configuration 20, but it was the one that got the highest score.

For configurations 14, 15, 21 and 22, the results were essentially copies of the 'bird' concept map, whereas 19 and 21 yielded highly unstable partial projections of both the 'horse' and the 'bird' concept maps simultaneously into the blend, since each of the 30 runs returned a different concept map. In the latter case, we find it difficult to interpret anything. A possible explanation for these unsuccessful configurations is that the frames used are much too abstract, leaving no concrete goal for the system.

Table 5.7 Parameters for configurations 11 to 22

Experiment	Weights					Query	Mapping
No.	I	T	MVR	U	R		
11	34	16	10	4	36	new_ability+aframe+aprojection	1
12	49	0	0	0	51	new_ability+aframe+aprojection	1
13	49	0	0	0	51	new_ability+ bprojection + bframe	1
14	34	16	10	4	36	new_ability + bprojection + bframe	1
15	34	16	10	4	36	bprojection + bframe	1
16	34	16	10	4	36	new_ability + bprojection + bframe	3
17	34	16	10	4	36	bprojection+ aframe	1
18	34	16	10	4	36	bprojection+ aframe	3
19	34	16	10	4	36	aprojection+ bframe	3
20	19	19	12	4	46	transport_means+bframe+bprojection	3
21	19	19	12	4	46	transport_means+bframe+bprojection	1
22	19	19	12	4	46	transport_means+bframe+bprojection	5

We also developed (in collaboration with Pablo Gervás (Pereira & Gervás, 2003)) an interpreter for generating textual descriptions of the blends, based on natural ganguage Generation. This system made descriptions by comparison with the input concepts (the initial descriptions of 'horse' and 'bird'). Examples of automatically generated descriptions of blends are:

(1) A horsebird is a horse. A horsebird has two wings and feathers. It can fly, and it moves by flying.

(2) A horsebird is a horse. A horsebird can fly, it has feathers, a beak, and wings for flying and it moves by flying.

(3) A horsebird is a horse. A horsebird can fly. It chirps, it has wings for flying and it moves by flying.

The example (1) corresponds to the Pegasus of configuration 9. Examples (2) and (3) are interpretations from configuration 7.

5.4.4 Recent Developments

Our research team was involved until recently in the Future and Emerging Technologies (FET) project ConCreTe (Concept Creation Technology).[2] This project was aimed at studying computational models for the representation and generation of previously unseen concepts and applying them in different domains such as narrative, poetry, or design (Cardoso, Martins, Assunção, Correia, & Machado, 2015).

The project revolved around a framework of a computational cognitive architecture that aimed to simulate human *spontaneous creativity*. The architecture combines ideas from Baars' global workspace theory (Baars, 1988), adopting principles from Peter Gärdenfors' theory of conceptual spaces (Gärdenfors, 2000) and from Shannon's information theory (Shannon, 1948). In this architecture, a number of concept generators perform their actions in parallel, and compete for access to a global workspace, where a threshold-based mechanism controls the access of the generators to the workspace.

Divago contributes to the framework as a (concept) generator. The inclusion of Divago in the architecture has been a motivation to work on new features of the system, namely the automatic selection of input spaces and more refined ways of selecting elements for projection.

One of the relevant outputs from the project was the integration of a version of Divago based on web services, called *DivagoFlow*, in a computational creativity infrastructure for online software composition, ConCreTeFlows[3] (Žnidaršič et al., 2016), which makes the framework available to the community.

Gonçalves, Martins, Cruz, and Cardoso (2015) presented some preliminary work on the automatic selection of the input spaces. These authors suggest an evolutionary algorithm, inspired by the work of Nagel, Thiel, Kötter, Piatek, and Berthold (2012) in bisociative knowledge discovery, for selecting two domains in the form of concept maps from a broader knowledge structure (Lavrač et al., 2019). More precisely, given a (large) semantic graph, the algorithm adopts a spreading activation technique to identify two partially overlapping subgraphs with similar sizes, while minimising overlap. A prototype of the algorithm was recently integrated and tested in the above-mentioned DivagoFlow framework, providing a means to automatically select the input spaces.

The Divago framework was also studied in the scope of case-based reasoning (CBR) (Cardoso & Martins, 2015). The idea behind this study was to analyse the potential role of CB in CBR, namely in the *Reuse* and *Revise* tasks of the classic '4 RES' model (Aamodt & Plaza, 1994), as an alternative mechanism that might provide better solutions in computational creativity set-ups. According to the proposed model, the case selected in the *Retrieve* task is blended with knowledge from a different domain. This may prove especially effective in computational creativity contexts, where it may provide an iterative divergence mechanism coupled with evaluation. The criteria for evaluating each possible blend may combine measures of coherence

[2] http://www.conceptcreationtechnology.eu

[3] http://concreteflows.ijs.si

with measures of distance from the given problem specification. In this context, the modules *Mapper* and *Blender* of Divago will be used with the aim of analysing the generation potential of two given domains. The GA-like search for blends is guided by an implementation of a variation of the optimality principles which favours the coherence of the resulting blends. In the scope of this work, an additional metric is required, as it is fundamental to measure the similarity to the original problem specification.

More recently, Gonçalves, Martins, and Cardoso (2017) re-implemented the process of search for the best blend, which in the original Divago is managed by the modules *Factory* and *Elaboration*, to handle it as an optimisation task by a multi-threaded genetic algorithm. The new module, called *Blendville*, also explores the use of multiple analogies in projecting concepts into the blend space and in the evolutionary process.

5.5 Conclusions

Interdisciplinary research, crossing efforts from cognitive science and artificial intelligence, has been contributing to multiple developments in computational creativity in recent decades. Since the seminal work of Goguen (1999), which introduced a formal theory of conceptual blending, various computational models of CB theory have been proposed and used in the design of creative systems.

In this chapter, we have revisited Divago, one of the first computational systems based on the CB mechanism. We have presented a thorough description of its main components and features, along with revisiting past publications that report work related to this architecture, particularly the work on concept creation.

Divago simulates the blending mechanism by exploring rule-based mechanisms, genetic algorithms and connectionism, among other techniques. One of is key features is the implementation of the optimality principles of CB theory. It is worth noting that several other existing computational models of CB do not explicitly implement this component.

Our more recent research deals with some of the limitations of the framework, which are also related to some of the current open issues in CB modelling. This includes the selection of input spaces, the selection of elements for projection and the definition of stopping criteria for the elaboration. We believe that creativity assessment can play an important part in this clarification, not merely as tool to evaluate artefacts, but as something that helps to understand what makes a good blend and how this translates into mechanisms such as the selection of input spaces or elements or the definition of the optimality principles.

The availability of the framework as a web service, as referred to in Section 5.4.4, opens up new possibilities for the use of Divago in wider contexts. An obstacle to such use, however, is the need for hand-crafted knowledge bases, given the current lack of suitable knowledge resources (e.g. concept maps) for providing adequate input spaces to the system. This seems to be an obstacle shared by other CB ap-

proaches, except for the proposal by Veale (2012), which addresses this issue by extracting knowledge from free online data that is available, although it is being used for a constrained version of a conceptual blend (conceptual mash-ups). However, the idea of extracting knowledge from online resources, although it is not new, shows a promising direction of research that we intend to explore.

Acknowledgements The authors acknowledge financial support from the FET programme within the Seventh Framework Programme for Research of the European Commission, under the ConCreTe FET-Open project (grant number 611733) and the PROSECCO FETProactive project (grant number 600653).

References

Aamodt, A., & Plaza, E. (1994). Case-based reasoning: Foundational issues, methodological variations, and system approaches. *AI Communications*, *7*(1), 39–59.

Baars, B. J. (1988). *A Cognitive Theory of Consciousness*. Cambridge University Press.

Besold, T. R., & Plaza, E. (2015). Generalize and Blend: Concept Blending Based on Generalization, Analogy, and Amalgams. In *Proceedings of the 6th International Conference on Computational Creativity, ICCC-15*.

Bou, F., Schorlemmer, M., Corneli, J., Gomez-Ramirez, D., Maclean, E., Smaill, A., & Pease, A. (2015). The role of blending in mathematical invention. In *Proceedings of the 6th International Conference on Computational Creativity, ICCC-15*.

Brandt, L., & Brandt, P. A. (2005). Making sense of a blend: A cognitive-semiotic approach to metaphor. *Annual Review of Cognitive Linguistics*, *3*(1), 216–249.

Cardoso, A., & Martins, P. (2015). Conceptual Blending in Case Adaptation. In *Proceedings of the 23rd International Conference on Case-Based Reasoning, ICCBR-15, CEUR Workshop Proceedings*.

Cardoso, A., Martins, P., Assunção, F., Correia, J., & Machado, P. (2015). A Distributed Approach to Computational Creativity. In *Proceedings of the 9th International Symposium on Intelligent Distributed Computing, IDC 2015*.

Confalonieri, R., Corneli, J., Pease, A., Plaza, E., & Schorlemmer, M. (2015). Using Argumentation to Evaluate Concept Blends in Combinatorial Creativity. In *Proceedings of the 6th International Conference on Computational Creativity, ICCC-15*.

Fauconnier, G. (1994). *Mental Spaces: Aspects of Meaning Construction in Natural Language*. New York: Cambridge University Press.

Fauconnier, G., & Turner, M. (1998). Conceptual Integration Networks. *Cognitive Science*, *22*(2), 133–187.

Fauconnier, G., & Turner, M. (2002). *The Way We Think*. New York: Basic Books.

Gärdenfors, P. (2000). *Conceptual Spaces: The Geometry of Thought*. Cambridge, MA: MIT Press.

Goguen, J. (1999). An Introduction to Algebraic Semiotics, with Applications to User Interface Design. In C. Nehaniv (Ed.), *Computation for Metaphor, Analogy and Agents, Lecture Notes in Artificial Intelligence* (Vol. 1562, pp. 242–291). Lecture Notes in Artificial Intelligence. Springer.

Goguen, J. (2005). What is a concept? In F. Dau, M.-L. Mugnier, & G. Stumme (Eds.), *Conceptual Structures: Common Semantics for Sharing Knowledge. Proceedings of the 13th International Conference on Conceptual Structures, ICCS 2005, Kassel, Germany, July 17–22, 2005* (Vol. 3596, pp. 52–77). Lecture Notes in Artificial Intelligence. Springer.

Goguen, J., & Malcom, G. (1996). *Algebraic Semantics of Imperative Programs*. MIT Press.

Gonçalves, J., Martins, P., & Cardoso, F. A. (2017). Blend City, BlendVille. In *Proceedings of the 8th International Conference on Computational Creativity, ICCC-17*, A.

Gonçalves, J., Martins, P., Cruz, A., & Cardoso, A. (2015). Seeking divisions of domains on semantic networks by evolutionary bridging. In *Workshop Proceedings of the 23rd International Conference on Case-Based Reasoning, ICCBR-15* (Vol. 1520, pp. 113–122).

Grady, J. E., Oakley, T., & Coulson, S. (1999). Blending and Metaphor. In R. W. Gibbs & G. Steen (Eds.), *Metaphor in Cognitive Linguistics*. Amsterdam: Benjamins.

Guhe, M., Pease, A., Smaill, A., Martinez, M., Schmidt, M., Gust, M., … Krumnack, U. (2011). A computational account of conceptual blending in basic mathematics. *Cognitive Systems Research*, *12*(3–4), 249–265.

Guilford, J. P. (1950). Creativity. *American Psychologist*, *5*(9), 444–454.

Gust, H., Kühnberger, K.-U., & Schmid, U. (2006). Metaphors and heuristic-driven theory projection (HDTP). *Theoretical Computer Science*, *354*(1), 98–117.

Johnson-Laird, P. N. (2002). How Jazz Musicians Improvise. *Music Perception*, *19*(3), 415–442.

Keane, M. T., & Costello, F. J. (2001). Setting limits on analogy: Why conceptual combination is not structural alignment. In D. Gentner, K. J. Holyoak, & B. Kokinov (Eds.), *The Analogical Mind: A Cognitive Science Perspective* (pp. 287–312). Cambridge, MA: MIT Press.

Koestler, A. (1964). *The Act of Creation*. New York: Macmillan.

Kutz, O., Neuhaus, F., Mossakowski, T., & Codescu, M. (2012). Blending in the Hub. In *Proceedings of the 5th International Conference on Computational Creativity, ICCC-14*.

Lakatos, I. (1976). *Proofs and Refutations: The Logic of Mathematical Discovery*. Cambridge University Press.

Lange, C., Kutz, O., Mossakowski, T., & Grüninger, M. (2012). The Distributed Ontology Language (DOL): Ontology Integration and Interoperability Applied to Mathematical Formalization. *CoRR*, *abs/1204.5*.

Lavrač, N., Juršič, M., Sluban, B., Perovšek, M., Urbančič, T., & Cestnik, B. (2019). Bisociative knowledge discovery for cross-domain literature mining. In T. Veale & F. A. Cardoso (Eds.), *Computational creativity: The philosophy and engineering of autonomously creative systems* (pp. 119–138). Springer.

Li, B., Zook, A., Davis, N., & Riedl, M. (2012). Goal-Driven Conceptual Blending: A Computational Approach for Creativity. In M. L. Maher, K. Hammond, A. Pease, R. Pérez, D. Ventura, & G. Wiggins (Eds.), *Proceedings of the 3rd International Conference on Computational Creativity, ICCC-12* (pp. 9–16). Dublin.

Martinez, M., Besold, T. R., & Abdel-Fattah, A. (2011). Towards a domain-independent computational framework for theory blending. In *Proceedings of the AAAI Fall Symposium on advances in cognitive systems* (pp. 210–217).

Martinez, M., Besold, T. R., Abdel-Fattah, A., Gust, H., Schmidt, M., Krumnack, U., & Kuehnberger, K.-U. (2012). Theory Blending as a Framework for Creativity in Systems for General Intelligence. In *Theoretical Foundations of Artificial General Intelligence* (pp. 219–239). Atlantis Press.

Nagel, U., Thiel, K., Kötter, T., Piatek, D., & Berthold, M. R. (2012). Towards Discovery of Subgraph Bisociations. In M. R. Berthold (Ed.), *Bisociative Knowledge Discovery* (Vol. 7250, pp. 263–284). Lecture Notes in Computer Science. Berlin: Springer.

Pereira, F. C. (2005). *Creativity and AI: A Conceptual Blending Approach* (Doctoral dissertation, University of Coimbra).

Pereira, F. C. (2007). *Creativity and Artificial Intelligence: A Conceptual Blending Approach*. Berlin: Mouton de Gruyter.

Pereira, F. C., & Cardoso, A. (1999). Dr. Divago: Searching for new ideas in a multi-domain environment. In *Proceedings of the 8th Cognitive Science of Natural Language Processing (CSNLP-8)*.

Pereira, F. C., & Cardoso, A. (2003). The horse-bird creature generation experiment. *AISB Journal*, *1*(3), 369.

Pereira, F. C., & Cardoso, A. (2006). Experiments with free concept generation in Divago. *Knowledge-Based Systems*, *19*(7), 459–471. doi:10.1016/j.knosys.2006.04.008

Pereira, F. C., & Gervás, P. (2003). Natural Language Generation from Concept Blends. In *AISB-03 Symposium on AI and Creativity in Arts and Science*.

Schorlemmer, M., Smaill, A., Kühnberger, K.-U., Kutz, O., Colton, S., Cambouropoulos, E., & Pease, A. (2014). COINVENT: Towards a Computational Concept Invention Theory. In *Proceedings of the 5th International Conference on Computational Creativity, ICCC-14*.

Schwering, A., Krumnack, U., Kuehnberger, K.-U., & Gust, H. (2009). Syntactic Principles of Heuristic-Driven Theory Projection. *Journal of Cognitive Systems Reseach*, *10*(3), 251–269.

Shannon, C. E. (1948). A Mathematical Theory of Communication. *Bell System Technical Journal*, *27*, 379–423.

Thagard, P., & Stewart, T. C. (2010). The AHA! Experience: Creativity Through Emergent Binding in Neural Networks. *Cognitive Science*, *35*(1), 1–33.

Veale, T. (1995). *Metaphor, Memory and Meaning: Symbolic and Connectionist Issues in Metaphor Interpretation*. PhD thesis, Dublin City University.

Veale, T. (2012). From Conceptual Mash-ups to Bad-ass Blends: A Robust Computational Model of Conceptual Blending. In *Proceedings of the 3rd International Conference on Computational Creativity, ICCC-12*.

Veale, T. (2019). From conceptual mash-ups to bad-ass blends: A robust computational model of conceptual blending. In T. Veale & F. A. Cardoso (Eds.), *Computational creativity: The philosophy and engineering of autonomously creative systems* (pp. 71–89). Springer.

Veale, T., & Keane, M. (1997). The Competence of Sub-Optimal Structure Mapping on "Hard" Analogies. In *Proceedings of ijcai'97, the 15th international joint conference on artificial intelligence*, San Mateo, California: Morgan Kaufmann.

Veale, T., & O'Donoghue, D. (2000). Computation and Blending. *Cognitive Linguistics*, *11*, 253–282.

Zacharakis, A., Kalikatso-Papakostas, M., & Cambouropoulos, E. (2015). Conceptual Blending in Music Cadences: A Formal Model and Subjective Evaluation. In *Proceedings of the 16th International Society for Music Information Retrieval (ISMIR) Conference*.

Żnidaršič, M., Cardoso, A., Gervás, P., Martins, P., Hervás, R., Alves, A. O., ... Lavrač, N. (2016). Computational Creativity Infrastructure for Online Software Composition: A Conceptual Blending Use Case. In *Proceedings of the 7th International Conference on Computational Creativity, ICCC-16*, Paris.

Chapter 6
Bisociative Knowledge Discovery for Cross-domain Literature Mining

Nada Lavrač, Matjaž Juršič, Borut Sluban, Matic Perovšek, Senja Pollak, Tanja Urbančič, and Bojan Cestnik

Abstract Given its immense growth, the scientific literature can be explored to reveal new discoveries, based on as yet undiscovered relations between knowledge from different, relatively isolated fields of specialization. This chapter presents an approach to creative knowledge discovery through the mechanism of *bisociation*. Bisociative reasoning is at the heart of creative, accidental discovery, i.e., serendipity. Bisociative knowledge discovery is focused on finding unexpected links by crossing between different contexts. In this work, bisociative knowledge discovery is explored in the framework of text mining, addressing cross-domain literature-based discovery. Two approaches are briefly outlined: the CrossBee approach to cross-domain bridging-term detection, and the OntoGen approach to bridging-term detection through outlier document exploration.

Nada Lavrač
Jožef Stefan Institute, Ljubljana, Jožef Stefan International Postgraduate School, Ljubljana and University of Nova Gorica, Slovenia. e-mail: nada.lavrac@ijs.si

Matjaž Juršič, Borut Sluban, Matic Perovšek
Jožef Stefan Institute, Ljubljana and Jožef Stefan International Postgraduate School, Ljubljana, Slovenia. e-mail: matjaz.jursic@ijs.si,borut.sluban@ijs.si,matic.perovsek@ijs.si

Senja Pollak
Jožef Stefan Institute, Ljubljana, Slovenia. e-mail: senja.pollak@ijs.si

Tanja Urbančič
University of Nova Gorica and Jožef Stefan Institute, Ljubljana, Slovenia.
e-mail: tanja.urbancic@ung.si

Bojan Cestnik
Temida d.o.o., Ljubljana and Jožef Stefan Institute, Ljubljana, Slovenia.
e-mail: bojan.cestnik@temida.si

T. Veale, F. A. Cardoso (eds.), *Computational Creativity*, Computational Synthesis and Creative Systems, https://doi.org/10.1007/978-3-319-43610-4_6

6.1 Introduction

The growing amounts of available knowledge and data exceed human analytic capabilities. Therefore new technologies that can help in analyzing and extracting useful information from large amounts of data need to be developed and used for analytic purposes. Understanding complex phenomena and solving difficult problems often requires knowledge from different domains to be combined and cross-domain associations to be considered. While the concept of association is at the heart of several information technologies, including information retrieval and data mining, and in particular association rule learning (Agrawal, Mannila, Srikant, Toivonen, Verkamo, et al., 1996), scientific discovery requires creative thinking to connect seemingly unrelated information, for example, by using metaphors or analogies between concepts from different domains. These kinds of context-crossing associations, called *bisociations* (Koestler, 1964), are often needed for innovative discoveries.

This chapter provides an introduction to bisociative knowledge discovery, and outlines selected approaches to cross-domain literature mining that support experts in searching for hidden links connecting two seemingly unrelated domains, most notably the CrossBee approach to cross-domain bridging-term (b-term) detection (Juršič, Cestnik, Urbančič, & Lavrač, 2012a, 2012b), and the approach to cross-domain literature mining via outlier document detection and exploration (Petrič, Cestnik, Lavrač, & Urbančič, 2012; Sluban, Juršič, Cestnik, & Lavrač, 2012).

This chapter is organized as follows. Section 6.2 presents related work in the area of bisociative knowledge discovery, literature-based discovery (LBD) and the human–computer interaction (HCI) aspects of creativity support tools. Section 6.3 illustrates the problem of b-term ranking and exploration through a use case scenario, followed by an overview of the b-term detection and exploration methodology as implemented in the CrossBee exploration tool, including the ensemble heuristic used in b-term detection. Section 6.4 presents two approaches to outlier document detection that can be used to narrow down the search space of b-terms, given the fact that outlier documents contain most of the cross-domain b-terms, as shown in our past research. Finally, Section 6.5 concludes with a summary of the methods presented and directions for further work.

6.2 Related Work

This section presents related work in the area of bisociative knowledge discovery, LBD and the HCI aspects of creativity support tools.

6.2.1 Bisociative Knowledge Discovery

Bisociative knowledge discovery is a challenging task motivated by a trend of over-specialization in research and development, which usually results in deep and relatively isolated silos of knowledge. Scientific literature too often remains closed, and cited only in professional subcommunities. Information that is related across different contexts is difficult to identify using associative approaches, like standard association rule learning (Agrawal et al., 1996) known from the data-mining and machine learning literature. Therefore, the ability of literature-mining methods and software tools to support experts in their knowledge discovery processes – especially in searching for yet unexplored connections between different domains – is becoming increasingly important.

Koestler (1964) argued that the essence of creativity lies in "perceiving of a situation or idea . . . in two self-consistent but habitually incompatible frames of reference," and introduced the expression *bisociation* to characterize this creative act. More specifically, Koestler's notion of *bisociation* was originally defined as follows:

> The pattern . . . is the perceiving of a situation or idea, L, in two self-consistent but habitually incompatible frames of reference, M_1 and M_2. The event L, in which the two intersect, is made to vibrate simultaneously on two different wavelengths, as it were. While this unusual situation lasts, L is not merely linked to one associative context but *bisociated* with two.

Koestler found bisociation to be the basis for human creativity in seemingly diverse human endeavors, such as humor, science, and the arts. As an example of bisociative scientific discovery, Koestler (p. 105) cites the "Eureka" discovery of Archimedes, bisociating the measurement of the volume of nonregular solids with the displacement of water:

> No doubt he had observed many times that the level of the [bath] water rose whenever he got into it; but this fact, and the distance between the two levels, was totally irrelevant to him – until it suddenly became bisociated with his problem. At that instant he realised that the amount of rise of the water-level was a simple measure of the volume of his own complicated body.

The concept of bisociation is illustrated in Fig. 6.1. It should be noted that context crossing is subjective, since the user has to move from their "normal"' context (frame of reference) to a *habitually incompatible context* to find the bisociative link. In Koestler's terms (Fig. 6.1), a habitual frame of reference (plane M_1) corresponds to the domain defined by the user. Other domains represents different, habitually incompatible contexts (in general, there may be several planes M_2). The creative act here is to find links (from S to the target T) which lead "out-of-the-plane" via intermediate, bridging concepts (L). Thus, contextualization and link discovery are two of the fundamental mechanisms in bisociative reasoning.

In summary, according to Koestler (1964), bisociative thinking occurs when a problem, idea, event, or situation is perceived simultaneously in two or more "matrices of thought" or domains. When two matrices of thought interact with each other, the result is either their fusion in a novel intellectual synthesis or their confrontation in a new esthetic experience. Koestler regarded many different mental phenomena

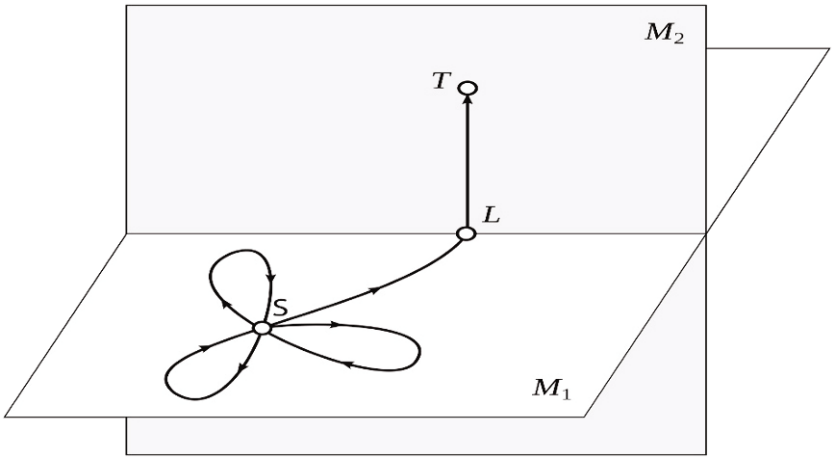


Fig. 6.1 Koestler's schema of bisociative discovery in science (Koestler, 1964, p. 107).

that are based on comparison (such as analogies, metaphors, jokes, identification, and anthropomorphism) as special cases of bisociation.

More recently, this work was followed by researchers interested in so-called *bisociative knowledge discovery*, where – according to Berthold (2012) – two concepts are bisociated if there is no direct, obvious evidence linking them and if one has to cross different domains to find the link, where a new link must provide some novel insight into the problem addressed. Bisociative knowledge discovery has become a topic of extensive research, addressing the discovery of bridging links or bridging concepts crossing between different domains and representations.

In modern terms (Berthold, 2012), bisociative knowledge discovery thus addresses a data-mining task where two or more domains of interest are searched for bisociative links or bridging concepts (i.e., individual context-bridging terms). Note that in this context, a single *domain* does not necessary refer to a single feature space; instead, we use this term to denote that the objects under analysis all represent properties with respect to one – more or less specific – aspect, even with multiple representations of the same space of objects (multiview learning, parallel universes, and redescription mining are well-known techniques addressing multiple representations of objects in the same domain of discourse). In contrast, bisociative knowledge discovery looks for connections between different domains of discourse, using either the same representation of different domains or different domain representations, where – according to Berthold (2012) – bridging concepts can be detected as nodes bridging different graphs, as subgraphs linking different graphs, as bridging links in terms of graph similarity, or as bridging terms appearing in different document corpora. The latter, referred to as *bridging-term discovery*, is the focus of the research described in this chapter.

6.2.2 Literature-Based Discovery

In LBD (Bruza & Weeber, 2008) – and, in particular, in cross-domain literature mining, which addresses knowledge discovery in two (or several) initially separate document corpora – a crucial step is the identification of interesting b-terms that carry the potential tp revealing the links connecting the separate domains. As shown by Petrič et al. (2012), LBD (Bruza & Weeber, 2008) is closely related to bisociative knowledge discovery (Berthold, 2012); for example, the b-terms known from the LBD literature directly correspond to Koestler's notion of bridging concepts *L*, introduced in the previous section.

The early work in LBD was due to Swanson (1990) and Smalheiser and Swanson (1998), who developed an approach to assist a user in LBD by detecting interesting cross-domain terms with the goal of uncovering the possible relations between previously unrelated concepts. The ARROWSMITH online system, developed by Smalheiser and Swanson (1998), takes as input two sets of titles of scientific papers from disjoint domains (disjoint document corpora) *A* and *C*, and lists terms that are common to *A* and *C*; the resulting b-terms are investigated further by the user for their potential to generate new scientific hypotheses.[1] Their approach, known as the "ABC model of knowledge discovery", addresses several settings, including the *closed discovery* setting (Weeber, Klein, de Jong-van den Berg, Vos, et al., 2001), where two initially separate domains *A* and *C* are specified by the user at the beginning of the discovery process, and the goal is to search for a bridging concept (term) *b* in *B* in order to support the validation of the hypothesized connection between *A* and *C*. The closed discovery setting, which is the most frequently addressed LBD setting, is illustrated in Fig. 6.2.

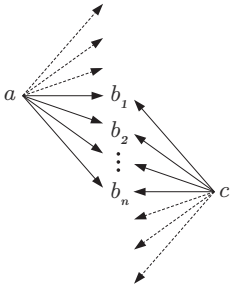


Fig. 6.2 Closed discovery process defined by Weeber, Klein, de Jong-van den Berg, Vos, et al. (2001).

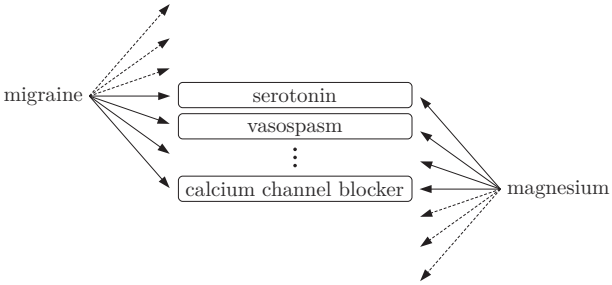


Fig. 6.3 Closed discovery when exploring migraine and magnesium documents, with b-terms identified by Swanson, Smalheiser, and Torvik (2006).

[1] In the ABC model, uppercase letter symbols *A*, *B*, and *C* are used to represent concepts (or sets of terms), and lowercase symbols *a*, *b*, and *c* to represent single terms.

Swanson's seminal work showed that databases such as PubMed can serve as a rich source of hidden relations between usually unrelated topics, potentially leading to novel insights and discoveries. By studying two separate literatures – the literature on migraine headache and the articles on magnesium – Swanson (1988) discovered "Eleven neglected connections", all of them supportive of the hypothesis that magnesium deficiency might cause migraine headache. Figure 6.3 illustrates the closed discovery setting for Swanson's task of finding the terms linking the "migraine" and "magnesium" domains. Swanson's literature-mining results were later confirmed by laboratory and clinical investigations. This well-known example has become the gold standard in the literature-mining field and has been used as a benchmark in several studies (Juršič et al., 2012b; Lindsay & Gordon, 1999; Srinivasan, 2004; Weeber et al., 2001).

Inspired by this early work, literature-mining approaches were developed further and successfully applied to different problems, such as finding associations between genes and diseases (Hristovski, Peterlin, Mitchell, & Humphrey, 2005), between diseases and chemicals (Yetisgen-Yildiz & Pratt, 2006), and others. Holzinger, Yildirim, Geier, and Simonic (2013) described several quality-oriented web-based tools for the analysis of the biomedical literature, which include analysis of terms (biomedical entities such as diseases, drugs, genes, proteins, and organs) and provide concepts associated with a given term. A more recent approach by Kastrin, Rindflesch, and Hristovski (2014) is complementary to the other LBD approaches, as it uses different similarity measures (such as common neighbors, the Jaccard index, and preferential attachment) for link prediction of implicit relationships in the Semantic MEDLINE network.

Supporting the user in effectively searching for b-terms provided a motivation for developing the CrossBee approach to b-term detection applicable in the closed discovery setting (Juršič et al., 2012b), implemented through ensemble-based term ranking, where an ensemble heuristic composed of six elementary heuristics was constructed for term evaluation. This approach is described in more detail in Section 6.3. Furthermore, the research conducted by Petrič et al. (2012) and Sluban et al. (2012) suggests that b-terms are more frequent in documents that are in some sense different from the majority of documents in a given domain. For example, Sluban et al. (2012) have shown that such documents, considered as outlier documents of their own domain, contain a substantially larger amount of bridging/linking terms than the regular nonoutlier documents. This approach, using the OntoGen tool (Fortuna, Grobelnik, & Mladenić, 2006), is described in more detail in Section 6.4.

In conclusion, let us summarize the relationship between bisociative knowledge discovery and Swanson's ABC model of literature-based discovery, where the particular focus of interest is the relationship between Koestler's bisociative link discovery framework and Weeber's closed discovery framework. Petrič et al. (2012) have presented a unifying view that establishes relationships between the two frameworks, as summarized in Table 6.1. Similarly to a bisociation, which, according to Koestler, is a result of processes of the mind when making new associations between concepts S and T from usually separated contexts (illustrated in Fig. 6.1), literature-based discoveries in Swanson's ABC model are a result of uncovering links between con-

Table 6.1 Unifying Koestler's and Swanson's models of creative knowledge discovery (Petrič, Cestnik, Lavrač, & Urbančič, 2012)

Koestler's model	Swanson's model
Bisociative link discovery process	Closed discovery process
Frames of reference (contexts) M_1 and M_2	Domains of interest A and C
Bisociative cross-context link $L \in M_1 \cap M_2$	Bridging term $b \in \text{terms}(A) \cap \text{terms}(C)$

cepts a and c from disjoint literatures A and C (illustrated in Fig. 6.2). In terms of Koestler's model, the two domains A and C, investigated in the closed LBD framework, correspond to the two habitually incompatible frames of reference, M_1 and M_2. Moreover, the b-terms $b_1, b_2, \ldots, b_n$ that are common to literatures A and C, clearly correspond to Koestler's notion of a situation or idea, L, which is not merely linked to one associative context but bisociated with two contexts M_1 and M_2.

6.2.3 Creativity Support Tools and HCI

CrossBee and the outlier detection tools developed can be viewed as creativity support tools, which are closely related to the field of HCI, as stated by Resnick et al. (2005) when summarizing the aims of designing creativity support tools (CSTs) as follows:

> Our goal is to develop improved software and user interfaces that empower users to be not only more productive, but more innovative.

The work of Shneiderman (2007, 2009) provides a structured set of design principles for CSTs, outlined below:

- *Support exploration*. To be successful in discovery and innovation, users should have access to improved search services providing rich mechanisms for organizing search results by ranking, clustering, and partitioning, with ample tools for annotation, tagging, and marking.
- *Enable collaboration*. While the actual discovery moments in innovation can be very personal, the processes that lead to them are often highly collaborative.
- *Provide rich history-keeping*. The benefits of rich history-keeping are that users have a record of which alternatives they have tried, they can compare the many alternatives, and they can go back to earlier alternatives to make modifications.
- *Design with low thresholds, high ceilings, and wide walls*. CSTs should have a short learning curve for novices (low threshold), yet provide sophisticated functionality that experts need (high ceilings), and also deliver a wide range of supplementary services to choose from (wide walls).

These principles were followed in our implementations and used in the evaluation of our approaches outlined in Sections 6.3 and 6.4 below, using two creativity support

tools CrossBee (Juršič et al., 2012a, 2012b) and OntoGen (Fortuna et al., 2006), respectively:

- *CrossBee* (Juršič et al., 2012a, 2012b) is an off-the-shelf solution for finding bisociations bridging two user-defined domains (separate domain literatures). CrossBee is a system that suggests b-terms using an ensemble-based term-ranking methodology. The tool also helps experts in searching for hidden links that connect two seemingly unrelated domains. In addition to this core functionality, supplementary functionality and content presentations have been added, which make the CrossBee web application a user-friendly tool for the ranking and exploration of prospective cross-context links. This enables the user not only to spot but also to efficiently investigate the cross-domain links discovered. The CrossBee user-friendly human–computer interface is briefly presented in Section 6.3.4.
- *OntoGen* (Fortuna et al., 2006) is a semiautomatic data-driven interactive text-mining tool that aids the user during the creative process of topic ontology construction. In essence, it is a text-mining tool for grouping documents into cohesive clusters, which can be considered as concepts in an automatically constructed topic ontology. The underlying methodology is k-means clustering, which is a particularly popular technique, since only the parameter k needs to be chosen to determine the number of categories in to which the documents will be clustered. A particularly appealing feature is OntoGen's user-friendly human–computer interface. The "main window" provides ontology visualization, where each concept is represented by the top three keywords (automatically assigned names of clusters, which can be manually edited), while the "concept hierarchy" window offers a quick overview of all the concepts with their positions in the concept hierarchy, which can also be directly manipulated. A particular use of OntoGen for outlier document detection is described in Section 6.4.2.

6.3 Bridging-Term Detection in Literature-Based Discovery

This section briefly describes our previous work on bisociative knowledge discovery in the area of literature mining, focusing on the CrossBee methodology for b-term detection, outlined in Juršič et al. (2012a, 2012b).

6.3.1 CrossBee Methodology

In cross-domain knowledge discovery, estimating which of the terms have a high potential for interesting discoveries is a challenging research question. It is especially important for cross-context scientific discovery such as understanding complex medical phenomena or finding new drugs for illnesses yet not fully understood.

The ensemble-based ranking methodology for b-term detection is illustrated in Fig. 6.4, showing the term ranking using an ensemble heuristic. Figure 6.5 shows the list of b-terms ranked by voting of an ensemble of heuristics, where the ranked list presented is the actual output produced by the CrossBee b-term exploration system using the gold standard dataset in literature mining, i.e., the combined migraine–magnesium dataset (Swanson, 1988). The ranked list of b-term candidates, shown in Figure 6.5, provides the user with some additional information, including the votes of the individual base heuristics in the ensemble and the domain occurrence statistics of the terms in both domains.

6.3.2 *Heuristics for Bridging-Term Discovery*

Several different elementary and ensemble heuristics for b-term ranking are available in CrossBee. The heuristics are defined as functions that numerically evaluate the quality of a term by assigning a bisociation score to it (measuring the potential that a term is actually a b-term). For the definition of an appropriate set of heuristics, we define a set of special (mainly statistical) properties of terms, which are aimed at distinguishing b-terms from regular terms; thus, these heuristics can also be viewed as advanced term statistics.

Formally, a heuristic is a function with two inputs, i.e., a set of domain-labeled documents D and a term t appearing in those documents, and one output, i.e, a score that represents the term's bisociation potential. All of the heuristics operate on data retrieved from the documents in text preprocessing. While term ranking using scores calculated by an ideal heuristic should result in ranking all the b-terms at the top of the ranked list, this ideal scenario is not realistic; nevertheless, ranking by heuristic scores should at least increase the proportion of b-terms at the top of the ranked term list.

We use the following notation: to state that a term's bisociation score b_score is equal to the result of a heuristic named $heurX$, we can write $b\_score = heurX(D,t)$. However, since the set of input documents is static when we are dealing with a

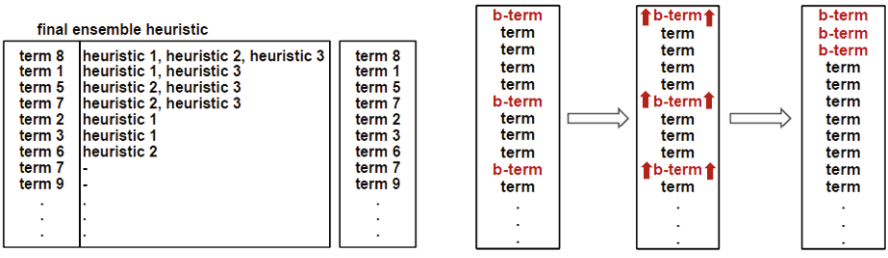


Fig. 6.4 Term-ranking approach: first, ensemble heuristics vote for terms, and next, terms are sorted according to their b-term potential (as shown on the left). Consequently, b-terms with the highest b-term potential should receive the highest scores (as shown on the right).

B-Term Identify (Analysis)

List start position: 0 Search terms(?): GO

There are 8058 documents in the database with 13445 terms (the termwhitelist contained 0 terms).
You have provided 43 bterms. Out of them 43 are found in the documents.

Pos.	Term	Votes	Inner Class Score	Documents		Hevristics' Votes			
				MIG	MAG	fr	td	cs	os
1	clinical	4	0,9935	115	88	X	X	X	X
2	therapy	4	0,9928	105	108	X	X	X	X
3	treatment	4	0,9923	362	156	X	X	X	X
4	trial	4	0,9915	73	16	X	X	X	X
5	case	4	0,9910	79	55	X	X	X	X
6	patient	4	0,9902	180	288	X	X	X	X
7	test	4	0,9902	17	37	X	X	X	X
8	syndrome	4	0,9901	45	44	X	X	X	X
9	magnesium	4	0,9899	1	5628	X	X	X	X
10	cerebral	4	0,9898	78	25	X	X	X	X
11	control	4	0,9898	68	70	X	X	X	X
12	drug	4	0,9896	75	32	X	X	X	X
13	pain	4	0,9894	33	5	X	X	X	X
14	study	4	0,9892	187	375	X	X	X	X
15	serotonin [1]	4	0,9892	63	8	X	X	X	X
16	artery	4	0,9891	41	24	X	X	X	X
17	prevention	4	0,9886	49	26	X	X	X	X
18	disease	4	0,9885	23	174	X	X	X	X
19	blood	4	0,9884	71	235	X	X	X	X
20	acid	4	0,9884	45	201	X	X	X	X

Fig. 6.5 The ensemble-heuristic-based ranking page, indicating with a cross (X) which elementary heuristics have identified the term as a potential b-term. This example shows the 20 top-ranked terms from the migraine–magnesium domain according to the selected heuristics.

concrete dataset, we can – for the sake of simplicity – omit the set of input documents from the notation for the heuristic and use $b\_score = heurX(t)$. Whenever we need to explicitly specify the set of documents to which a function is applied (this is never needed for a heuristic, but sometimes needed for auxiliary functions used in the formula for the heuristic), we write it as $funcX_D(t)$. To specify the function's input document set, we have two options: we can either use D_u, which stands for the (union) set of all the documents from all the domains, or use $D_n : n \in \{1..N\}$, which stands for the set of documents from the given domain n. In general, the following statement holds: $D_u = \cup_{n=1}^{N} D_n$, where N is the number of domains. In the most common scenario, when there are exactly two distinct domains, we also use the notation D_A for D_1 and D_C for D_2, similarly to Swanson's notation using the symbols A and C as representatives of the initial and the target domain in the closed discovery setting, as mentioned in Section 6.2.

We defined four sets of base heuristics: six frequency-based, four TF-IDF-weight-based ("TF-IDF" denotes the product of term frequency and inverse document frequency weights, frequently used in document vector representations in text mining (Salton & Buckley, 1988)), three similarity-based, and eight outlier-based heuristics. Most of the heuristics that we developed work in a fundamentally similar way – they all manipulate solely the data present in the term and document vectors and derive the bisociation score of the terms. The exceptions to this are the outlier-based heuristics, which first evaluate outlier documents and only later use the information from the

term vectors for b-term evaluation. Using these base heurisctics, we developed the ensemble heuristic described below.

6.3.3 Ensemble Heuristic

The ensemble heuristic for b-term discovery, which we constructed based on the experiments, is constructed as a sum of two parts, $s_t = s_t^{\text{vote}} + s_t^{\text{pos}}$, i.e., the ensemble voting score s_t^{vote} and the ensemble position score s_t^{pos}, which are summed together to give the final ensemble score for every term in the corpus vocabulary. Each term score represents the term's potential for linking the two disjoint domains.

The ensemble voting score (s_t^{vote}) of a given term t is an integer, which denotes how many base heuristics voted for the term: each term can be given a score $s_{t_j}^{\text{vote}} \in \{0,1,2,\ldots,k\}$, where k is the number of base heuristics used in the ensemble. The ensemble voting score of term t_j at position p_j in the ranked list of n terms is computed as a sum of the voting scores of the individual heuristics:

$$s_{t_j}^{\text{vote}} = \sum_{i=1}^{k} s_{t_j,h_i}^{\text{vote}} = \sum_{i=1}^{k} \begin{cases} 1, & p_j < n/3, \\ 0, & \text{otherwise.} \end{cases} \tag{6.1}$$

The ensemble position score (s_t^{pos}) is calculated as an average of the position scores of the individual base heuristics. For each heuristic h_i, the term's position score s_{t_j,h_i}^{pos} is calculated as $n - p_j/n$, which results in the position scores being in the interval $[0,1)$. For an ensemble of k heuristics, the ensemble position score is computed as an average of the position scores of the individual heuristics:

$$s_{t_j}^{\text{pos}} = \frac{1}{k} \sum_{i=1}^{k} s_{t_j,h_i}^{\text{pos}} = \frac{1}{k} \sum_{i=1}^{k} \frac{n - p_j}{n}. \tag{6.2}$$

The method of constructing the ensemble score described above looks rather intricate; however, the calculation of the ensemble score by our method is well justified by extensive experimental results (Juršič et al., 2012a, 2012b) on the migraine–magnesium dataset (Swanson, 1988). Based on the experimental results, the final set of elementary heuristics included in the ensemble consisted of the following heuristics:

- outFreqRelRF, the relative frequency of term t in the outlier document set detected by a random forest classifier;
- outFreqRelSVM, the relative frequency of term t in the outlier document set detected by a support vector machine classifier;
- outFreqRelCS, the relative frequency of term t in the outlier document set detected by a centroid similarity classifier;
- outFreqSum, the sum of the frequencies of term t in all three outlier document sets;

- tfidfDomnSum, the sum of the TF-IDF weights of term t in the two domains; and
- freqRatio, the term-to-document frequency ratio.

A detailed justification for the choice of this particular combination of heuristics is presented in Juršič (2015).

6.3.4 The CrossBee HCI Interface

The user-friendly CrossBee web interface can be used to efficiently investigate cross-domain links ranked by the ensemble-based ranking methodology. CrossBee's document-focused exploration empowers the user to filter and order the documents by various criteria, including a detailed document view that provides a more detailed presentation of a single document, including various term statistics. Methodology performance analysis supports the evaluation of the methodology by providing various data which can be used to measure the quality of the results, for example data for plotting ROC curves. High-ranked-term emphasis marks the terms according to their bisociation score calculated by the ensemble heuristic. When this feature is used, all high-ranked terms are emphasized throughout the whole application, thus making them easier to spot (see the different font sizes in Fig. 6.6). B-term emphasis marks the terms defined as b-terms by the user (terms highlighted in yellow in Fig. 6.6). Domain separation is a simple but effective option which colors all documents from the same domain in the same color, making an obvious distinction between the documents from the two domains (different colors in Fig. 6.6). User interface customization enables the user to decrease or increase the intensity of the following features: high-ranked term emphasis, b-term emphasis, and domain separation.

The user can inspect the actual appearances of the selected term in both domains, using side-by-side document inspection as shown in Fig. 6.6. In this way, they can verify whether their rationale behind selecting this term as a b-term can be justified based on the contents of the documents inspected.

6.4 Exploring Outlier Documents in Literature-Based Discovery

This section outlines the exploration of outlier documents as means for cross-domain LBD (Petrič et al., 2012; Sluban et al., 2012).Here, we use the term "outlier detection" to refer to the task of finding irregular or unusual data instances (documents in the case of literature mining) that do not conform to the expected distribution.

Outlier detection is an established area of data mining (Aggarwal, 2013). Conceptually, an outlier is an unexpected event or entity, or – in our case – an irregular document. We are especially interested in outlier documents since they frequently

embody new information that is hard to explain in the context of existing knowledge. Moreover, in data mining, an outlier is occasionally a primary object of study, as it can potentially lead to the discovery of new knowledge. These assumptions are well aligned with the bisociation potential that we wish to optimize; thus, we have constructed several heuristics that harvest the information possibly residing in outlier documents.

6.4.1 *Outlier Document Detection and B-term Identification Through Document Classification*

The technique proposed by Sluban et al. (2012) to detect outlier documents using classification algorithms, works as follows. Having documents from two domains of interest, we first train a classification model that distinguishes between the documents from these domains. Using the model constructed we classify all the documents. The documents that are misclassified – according to their domain of origin – are declared to be outlier documents, since according to the classification model they do not belong to their domain of origin. These domain outliers are actually borderline documents, as they were considered by the model to be more similar to the other domain than their originating domain. Hence they can be regarded as bridging documents between the two domains.

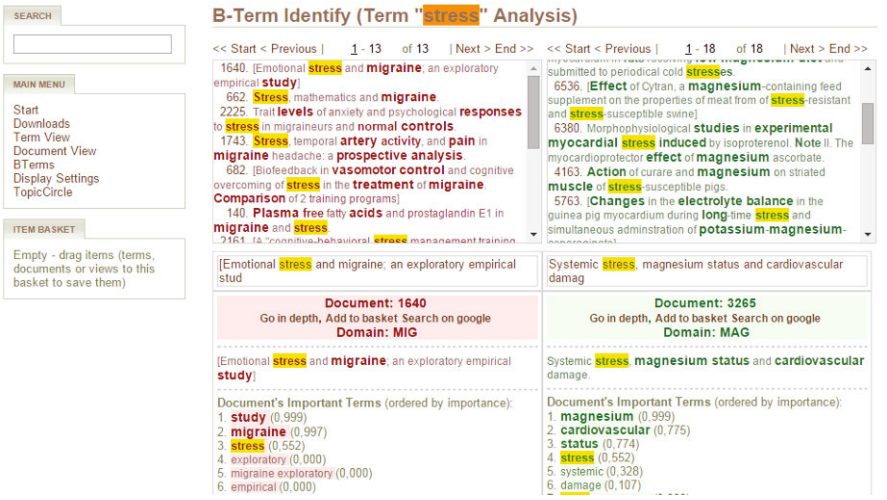


Fig. 6.6 One of the useful features of the CrossBee interface is the side-by-side view of documents from the two domains under investigation. The analysis of the b-term "stress" from the migraine–magnesium domain is shown. The view presented enables efficient comparison of two documents, the left one from the migraine domain and the right one from the magnesium domain.

In our work, we thus used noise detection approaches to find outlier documents containing cross-domain b-terms between two different domains. When exploring a domain pair dataset, we searched for a set of outlier documents using different classification-noise-filtering approaches (Brodley & Friedl, 1999), implemented and adapted for this purpose.

Classification noise filtering is based on the idea of using a classifier as a tool for detecting noisy and outlier instances in data. In this work, the simple classifiers used by Brodley and Friedl (1999) were replaced by new, better-performing classifiers, as the noise filter should, as much as possible, trust the classifiers that they will be able to correctly predict the class of a data instance. In this way, the incorrectly classified instances are considered to be noise/outliers. In other words, if an instance of class *A* is classified in the opposite class *C*, we consider it to be an outlier of domain *A*, and vice versa. We denote the two sets of domain outlier documents by $\mathcal{O}(\mathrm{A})$ and $\mathcal{O}(\mathrm{C})$, respectively. Figure 6.7 illustrates the principle.

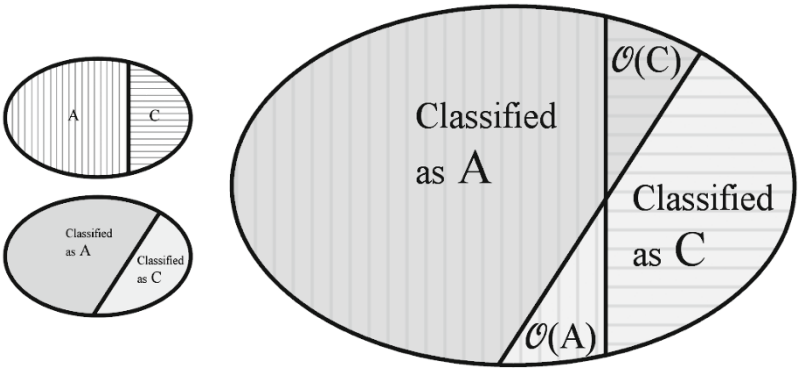


Fig. 6.7 Detecting outliers of a domain pair dataset using document classification.

We evaluated whether domain outliers obtained by classification noise filtering have the potential for bridging different concepts. We tested this on the migraine–magnesium (Swanson, Smalheiser, & Torvik, 2006) and autism–calcineurin (Petrič, Urbančič, Cestnik, & Macedoni-Lukšič, 2009) domain pair datasets, which have lists of confirmed concept b-terms. The experimental results showed that the sets of detected outlier documents were relatively small – including less than 5% of the entire datasets – and that they contained a great majority of b-terms; the number of b-terms in them was significantly higher than in same-sized random subsets. These results are summarized in Fig. 6.8. Hence the effort needed for finding cross-domain links is substantially reduced, as it requires one to explore a much smaller subset of documents, where a great majority of the b-terms are present and these terms are more frequent.

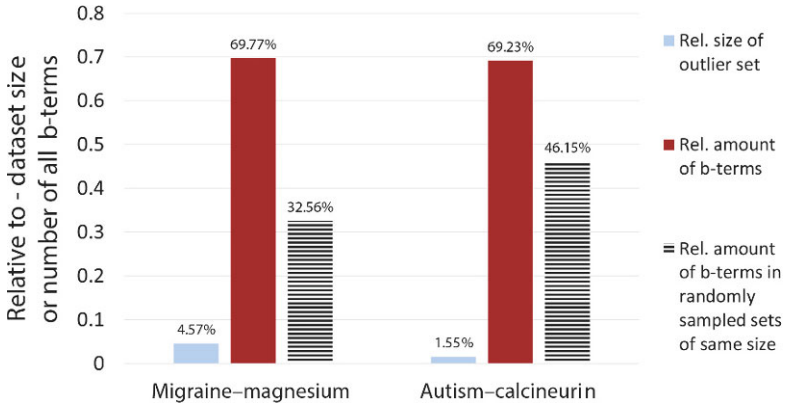


Fig. 6.8 Presence of b-terms in the detected outlier sets of two domain pair datasets.

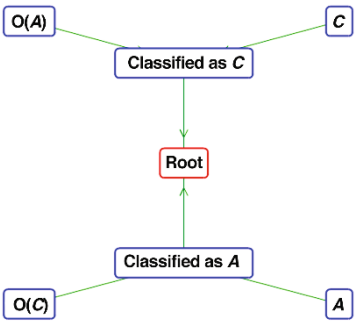


Fig. 6.9 Target domain documents from literatures A and C, clustered according to OntoGen's two-step approach, using first unsupervised and then supervised clustering to obtain outlier documents $\mathcal{O}(\mathrm{A})$ and $\mathcal{O}(\mathrm{C})$ of literatures A and C, respectively.

6.4.2 *Outlier Document Detection and B-term Identification Through Document Clustering*

The approach proposed by Petrič et al. (2012) concentrates on a specific type of outlier – the domain outliers – i.e., the documents that tend to be more similar to the documents in the opposite domain than to those in their own domain. In this approach, document clustering is used to find outlier documents. The approach consists of two steps. In the first step, the OntoGen clustering algorithm (Fortuna et al., 2006) is applied to cluster the merged document set $A \cup C$, consisting of documents from both of the domains A and C. The result of unsupervised clustering is two document clusters: A' = *Classified as A* (i.e., documents from $A \cup C$ classified as A), and C' = *Classified as C* (i.e., documents from $A \cup C$ classified as C). Then, in the second step, for each of the clusters, a supervised clustering approach is applied taking into account the original domains A and C of the documents. As a result, a two-level tree hierarchy of clusters is generated. The approach is illustrated in Fig. 6.9.

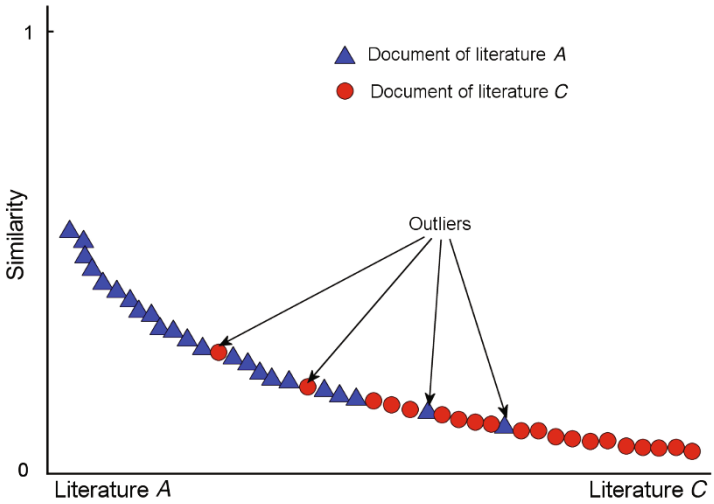


Fig. 6.10 Graph representing instances (documents) of literature *A* and instances (documents) of literature *C* according to their content similarity to a prototypical document of literature *A*, as suggested in Petrič, Cestnik, Lavrač, and Urbančič (2012). In this graph, outliers of literature *C* are positioned closer to the typical representatives of literature *A* than to the central documents of literature *C*.

The experimental results obtained in the gold standard migraine–magnesium domain, as well as in the autism–calcineurin domain pair, confirm the hypothesis that most b-terms appear in outlier documents and that, by considering only outlier documents, the search space for b-term identification can be greatly reduced. Moreover, the user can employ the document similarity graph – schematically presented in Fig. 6.10 – to identify the most irregular documents in their own domain and start the search for b-terms from these outlier documents, belonging to subclusters $\mathscr{O}(\mathrm{C})$ and $\mathscr{O}(\mathrm{A})$. In this way, the search space for finding b-term candidates can be substantially reduced.

6.4.3 Relating Outlier Document Detection to CrossBee Heuristics

The outlier document detection approaches described above inspired the development of outlier-based heuristics for the CrossBee b-term detection engine. As mentioned in Section 6.3.3, six heuristics (outFreqRelRF, outFreqRelSVM, outFreqRelCS, outFreqSum, tfidfDomnSum, and freqRatio) are used in the CrossBee ensemble heuristic. The outlier-based heuristics proved to be very effective. Note that four of these (outFreqRelRF, outFreqRelSVM, outFreqRelCS, and outFreqSum) are based on term frequencies in outlier documents; three of them were inspired by the classification-based approach (outFreqRelRF, outFreqRelSVM, and outFreqRelCS) and one by the OntoGen clustering approach (outFreqSum) to outlier document detection.

6.5 Conclusions and Further Work

This chapter has presented selected information technologies for creative knowledge discovery, developed to uncover previously unknown links between facts in different contexts, potentially leading to new insights and new knowledge. The approaches described are based on the Koestler's notion of bisociations, connecting domains that are usually considered as separate. When the domains investigated are described by texts, for example a set of documents, bisociative literature-mining methods can point towards novel chains of thoughts by identifying bridging terms with high potential for new discoveries resulting from putting existing pieces of knowledge together into a novel, interesting and reasonable whole.

The identification of cross-context links or bridging terms leading to new insights and discoveries is not an easy task, owing to the huge search space of possibilities, similar to looking for a needle in a haystack. One of the possible solutions is to identify the parts of the search space with an increased probability of finding good candidate terms/concepts with, the aim of restricting the huge amount of existing literature to a more manageable amount of sources to be explored first. This is the approach taken in our research in cross-domain literature mining via outlier document detection and exploration, presented in Section 6.4. The other option is to estimate the potential of candidate links for new discoveries and to concentrate on the most promising ones. This is the approach taken in the development of the online CrossBee application, supporting the user in the search and detection of cross-domain bridging terms, outlined in Section 6.3. The information technologies outlined, and other related approaches to bisociative link discovery that help in uncovering new connections between existing pieces of knowledge in the literature, can be used to assist researchers in their creative process by suggesting and even ranking candidate bridging terms.

Note that IT tools that implement literature-based discovery to enable a researcher to guide the discovery process by using his or her background knowledge enable the researcher to explore the literature more efficiently, but may also trigger the researcher's own human creativity. IT-supported literature exploration may provoke the researcher's own 'Koestler-style' bisociations to be triggered in this process. These bisociations may better specify or redirect the focus of further steps in the literature-based discovery process. The history of science and engineering offers numerous examples showing that Koestler's bisociative principle of thought has been an important element of new discoveries, based on innovative connections between already known ideas. As described above, literature-based discovery and bisociative knowledge discovery can complement each other, offering immense possibilities for new discoveries that we have only started to explore, but have already seen this process working at its best.

In future work we will introduce additional user interface options for data visualization and exploration as well as advance the term ranking methodology by adding new sophisticated heuristics, which will take into account also the semantic aspects of the data. Besides, we will apply the system to new domain pairs to exhibit its generality, investigate the need and possibilities of dealing with domain

specific background knowledge, and assist researchers in different disciplines in their explorations which may lead to new scientific discoveries. We will also propose a further extension of the literature based discovery methodology by facilitating the use of controlled vocabularies, enhancing the heuristics capability to rank the actual b-terms at the top of the ranked term list.

Acknowledgements This work was supported by the Slovenian Research Agency and the FP7 European Commission FET project PROSECCO (grant 600653).

References

Aggarwal, C. (2013). *Outlier analysis*. Springer.

Agrawal, R., Mannila, H., Srikant, R., Toivonen, H., Verkamo, A. I., et al. (1996). Fast discovery of association rules. *Advances in Knowledge Discovery and Data Mining*, *12*(1), 307–328.

Berthold, M. (Ed.). (2012). *Bisociative knowledge discovery*. Springer.

Brodley, C. E., & Friedl, M. A. (1999). Identifying mislabeled training data. *Journal of Artificial Intelligence Research*, *11*, 131–167.

Bruza, P., & Weeber, M. (2008). *Literature-based discovery*. Springer Science & Business Media.

Fortuna, B., Grobelnik, M., & Mladenić, D. (2006). Semi-automatic data-driven ontology construction system. In *Proceedings of the 9th International Multi-conference Information Society* (pp. 223–226).

Holzinger, A., Yildirim, P., Geier, M., & Simonic, K.-M. (2013). Quality-based knowledge discovery from medical text on the web. In G. Pasi, G. Bordogna, & L. C. Jain (Eds.), *Quality issues in the management of web information* (pp. 11–13). Springer.

Hristovski, D., Peterlin, B., Mitchell, J. A., & Humphrey, S. M. (2005). Using literature-based discovery to identify disease candidate genes. *International Journal of Medical Informatics*, *74*(2), 289–298.

Juršič, M. (2015). *Text mining for cross-domain knowledge discovery* (Doctoral dissertation, Jožef Stefan International Postgraduate School, Ljubljana, Slovenia).

Juršič, M., Cestnik, B., Urbančič, T., & Lavrač, N. (2012a). Bisociative literature mining by ensemble heuristics. In M. Berthold (Ed.), *Bisociative knowledge discovery* (pp. 338–358). Springer.

Juršič, M., Cestnik, B., Urbančič, T., & Lavrač, N. (2012b). Cross-domain literature mining: Finding bridging concepts with CrossBee. In *Proceedings of the 3rd international conference on computational creativity* (pp. 33–40).

Kastrin, A., Rindflesch, T. C., & Hristovski, D. (2014). Link prediction on the semantic MEDLINE network. In S. Džeroski, P. Panov, D. Kocev, & L. Todorovski (Eds.), *Discovery science* (pp. 135–143). Springer.

Koestler, A. (1964). *The act of creation*. Hutchinson.

Lindsay, R. K., & Gordon, M. D. (1999). Literature-based discovery by lexical statistics. *Journal of the American Society for Information Science and Technology*, *50*(1), 574–587.

Petrič, I., Cestnik, B., Lavrač, N., & Urbančič, T. (2012). Outlier detection in cross-context link discovery for creative literature mining. *Computer Journal*, *55*(1), 47–61.

Petrič, I., Urbančič, T., Cestnik, B., & Macedoni-Lukšič, M. (2009). Literature mining method RaJoLink for uncovering relations between biomedical concepts. *Journal of Biomedical Informatics*, *42*(2), 219–227.

Resnick, M., Myers, B., Nakakoji, K., Shneiderman, B., Pausch, R., Selker, T., & Eisenberg, M. (2005). Design principles for tools to support creative thinking. In *Proceedings of the nsf workshop on creativity support tools* (pp. 25–36).

Salton, G., & Buckley, C. (1988). Term-weighting approaches in automatic text retrieval. *Information Processing & Management*, *24*(5), 513–523.

Shneiderman, B. (2007). Creativity support tools: Accelerating discovery and innovation. *Communications of the ACM*, *50*(12), 20–32.

Shneiderman, B. (2009). Creativity support tools: A grand challenge for HCI researchers. In M. Redondo, C. Bravo, & M. Ortega (Eds.), *Engineering the user interface: From research to pratice* (pp. 1–9). Springer.

Sluban, B., Juršič, M., Cestnik, B., & Lavrač, N. (2012). Exploring the power of outliers for cross-domain literature mining. In M. Berthold (Ed.), *Bisociative knowledge discovery* (pp. 325–337). Springer.

Smalheiser, N., & Swanson, D. R. (1998). Using ARROWSMITH: A computer-assisted approach to formulating and assessing scientific hypotheses. *Computer Methods and Programs in Biomedicine*, *57*(3), 149–154.

Srinivasan, P. (2004). Text mining: Generating hypotheses from MEDLINE. *Journal of the American Society for Information Science and Technology*, *55*(5), 396–413.

Swanson, D. R. (1988). Migraine and magnesium: Eleven neglected connections. *Perspectives in Biology and Medicine*, *78*(1), 526–557.

Swanson, D. R. (1990). Medical literature as a potential source of new knowledge. *Bulletin of the Medical Library Association*, *78*(1), 29.

Swanson, D. R., Smalheiser, N. R., & Torvik, V. I. (2006). Ranking indirect connections in literature-based discovery: The role of medical subject headings (MeSH). *Journal of the American Society for Information Science and Technology*, *57*(11), 1427–1439.

Weeber, M., Klein, H., de Jong-van den Berg, L., Vos, R., et al. (2001). Using concepts in literature-based discovery: Simulating Swanson's Raynaud–fish oil and migraine–magnesium discoveries. *Journal of the American Society for Information Science and Technology*, *52*(7), 548–557.

Yetisgen-Yildiz, M., & Pratt, W. (2006). Using statistical and knowledge-based approaches for literature-based discovery. *Journal of Biomedical Informatics*, *39*(6), 600–611.

Chapter 7
Computational Design, Analogy, and Creativity

Ashok Goel

Abstract We first characterize creativity in design from an information-processing perspective and analyze analogical thinking as a core process of design creativity. Then, we describe a computational theory of analogical design called model-based analogy. Next, we summarize empirical results from in situ observations of analogical thinking in biologically inspired design. Then, we analyze biologically inspired design from the viewpoint of model-based analogy. Finally, we briefly reflect on computational creativity research on analogical design.

7.1 Introduction

Computational creativity has been called the "final frontier" of artificial intelligence (Colton & Wiggins, 2012). As this volume attests, much of the research on computational creativity pertains to art, literature, and entertainment, for example music, drawing and painting, story understanding and telling, poetry, limericks and puns, dance, drama, games, and so on. Of course, design too can be and often is creative. In this chapter, we are mostly interested in creativity in design.

However, design itself is very wide-ranging and open-ended, spanning domains from food to fashion design, from product to industrial design, from architecture to urban design, from engineering to system design, from organization to business design, etc. In this chapter, we focus on creativity in system design, that is, the design of artifacts that contain multiple interacting components and processes at multiple levels of abstraction.

Examples of creativity in system design are too numerous to enumerate here: an early example that revolutionized human civilization is the design of the wheel; a

Ashok Goel
Design & Intelligence Laboratory, School of Interactive Computing,
Georgia Institute of Technology, Atlanta, GA, USA
e-mail: goel@cc.gatech.edu

T. Veale, F. A. Cardoso (eds.), *Computational Creativity*, Computational Synthesis and Creative Systems, https://doi.org/10.1007/978-3-319-43610-4_7

more recent revolutionary example is the design of the computer. The economic and social impact of system design often is so transformative that we sometimes define human eras in its terms, such as the era of the internal combustion engine, the era of assembly line production of automobiles, and, now, the new era of self-driving cars. Thus, it is important to study creativity in system design as well as to build computational theories, techniques, and tools for supporting it in practice.

Creativity in system design is typically manifested in the early, conceptual phases engaging problem framing, idea generation and evaluation, problem reformulation, etc. However, much research on computer-aided design focuses on the latter phases of design such as geometric modeling, numerical analysis, simulation, and optimization. Thus, while creativity in systems design is an important area of research because of its potential for economic and social impact, it has received relatively little attention in the field of either computer-aided design or computational creativity.

The conceptual phase of system design takes a specification of a design problem as input. The design problem always specifies the function(s) desired of the system and may also specify the operating environment, performance criteria, and constraints on the system structure (Dym & Brown, 2012). The desired output of the conceptual phase of system design is a specification of the structure of the system such that the structure achieves the desired function(s) in the operating environment while also satisfying the specified constraints.

The conceptual phase of system design typically engages several fundamental processes of creativity, including (1) *design thinking*, or thinking about ill-structured, open-ended problems with ill-defined goals and evaluation criteria (Cross, 2011); (2) *systems thinking*, or thinking about complex phenomena consisting of multiple interacting components and causal processes (Forrester, 1961); (3) *analogical thinking*, or thinking about novel situations in terms of similar, familiar situations (K. Holyoak & Thagard, 1996); (4) *visual thinking* or thinking about images and in images (Arnheim, 1969); (5) *abductive thinking*, or thinking about possible explanations for data (Josephson & Josephson, 1996); and (6) *meta-thinking*, or thinking about one's own knowledge and thinking (Cox & Raja, 2011). In this chapter, we are mostly interested in analogical thinking.

7.2 Creativity in Design

Characterizations of creativity in general typically focus on traits of the product such as novelty, value, nonobviousness, or unexpectedness (Sternberg, 1999). In contrast, characterizations of creativity in design often emphasize design process and knowledge. These characterizations tend to be of two kinds. The first begins with a general theory of human information processing and then delimits routine, innovative, and creative design in the terms of the theory. For example, Newell and Simon (1972) describe a general theory of human problem solving. According to their theory, humans address problems by searching in problem spaces, where a problem space is abstractly defined by the goal and the domain knowledge, in the

form of operators that enable the search. When this theory is applied to design, the goals are part of the design requirements that express design variables and the ranges of values they can take; the operators are specific to the particular design domain. Some design theorists have characterized design creativity in terms of problem-space search: If the design variables and the ranges of values they can take remain fixed during design problem solving, the design is routine; if the design variables remain fixed but the ranges of values change, the design is innovative; and if the design variables and the range of values both change, the design is creative.

The second kind of characterization of creativity in design arises from specific theories of design. The design theorist posits a computational process for design, and then delimits routine, innovative, and creative design in the terms of the theory. For example, Brown and Chandrasekaran (1989) describe a computational theory of design based on the notion of design plans. This theory uses a structure–substructure hierarchy to organize skeletal design plans for the object to be designed. Designing a particular instance of the object involves selecting, instantiating, and expanding the design plans. Brown (1996) proposes a characterization of creative design based on this theory: if the designer knows both the structure of the design space (the structure–substructure hierarchy of the object) and procedures for systematically searching the space (the skeletal design plans), the design is routine; if the designer knows only the structure of the design space, the design is innovative; and if the designer knows neither, the design is creative.

Three implications follow directly from these two characterizations of design creativity. The first characterization suggests that problem formulation and reformulation are integral parts of creative design. A designer's understanding of a problem typically evolves during creative design processing. This evolution of understanding of the problem can lead to (possibly radical) changes in the problem and solution representations. The second characterization suggests that in creative design, knowledge needed to address a problem is typically not available in a form directly applicable to the problem. Instead, at least some of the needed knowledge must be acquired from other knowledge sources (by analogical transfer from a different problem, for example). Both characterizations indicate that creativity in design lies on a continuum. That is, creativity in design can occur in varying degrees, where the degree of creativity depends on the extent of problem and solution reformulation and the (analogical) transfer of knowledge from different knowledge sources to the design problem.

7.3 Analogical Thinking in Creative Design

My research on design since about the mid–1980s has explored analogy as a fundamental process of creativity. Let us begin with the traditional definition of analogy (Thagard, 2005): given a problem P_{new} and a (partial, possibly null) solution S_{new} to P_{new}, analogical reasoning involves retrieval of a familiar problem P_{old} from memory with a solution S_{old}, and transfer of selected elements from S_{old} to S_{new}. Thus, ana-

logical design involves the recall and transfer of elements of a solution to one design problem to a solution to another design problem, where the selected design elements can be components, relations between components, or configurations of components and relations.

It is productive to ask four questions in analyzing AI theories of creative design: *why*, *what*, *how*, and *when* (Goel, 1997). In the context of analogical design, the *why* question pertains to the task (or the goal) for which analogy is used – for example, the task of proposing a candidate design. The *what* question pertains to the content of the knowledge that is transferred – for example, the transfer of knowledge of a mechanism is to achieve a function from one design situation to another. The *how* question pertains to the methods for recall and transfer, where a method uses specific kinds of knowledge representations, inference mechanisms, and control strategies. Finally, the *when* question pertains to strategic control of the processing. Clearly, the four questions are related.

If we ask these questions about the traditional characterization of analogy in the context of design creativity, the characterization appears too narrow, in terms of both what can be transferred and why it can be transferred. (This characterization also implies that P_{old} and P_{new} are different design problems. But, in general, P_{old} and P_{new} can be subproblems of the same design problem. This is the familiar distinction between cross-problem and within-problem analogies).

Why? The traditional characterization of analogy implies that analogical recall and transfer occur in the service of generating the solution S_{new} to the design problem P_{new} – that is, the service of proposing a candidate design, modifying an initial design, and completing a partial design. But there is no principled reason to limit analogies only to these design tasks. Design, especially creative design, involves a variety of other design tasks, such as interpreting and elaborating the design problem, decomposing the problem, anticipating potential difficulties with a candidate solution, refining a candidate design, evaluating a candidate design, interpreting the evaluation information, and reformulating the problem. Analogies, in general, can help address any of these design tasks.

What? The answer depends in part on the design task being addressed. The earlier characterization of analogy implies that the content of knowledge transfer is in the form of design elements, for example, components and relations between components. This kind of knowledge transfer seems appropriate for the tasks of design proposition, modification, and completion. For other design tasks, however, analogies can result in the transfer of different kinds of knowledge, for example, transfer of knowledge of familiar design problems for the tasks of interpreting and reinterpreting a new problem, and transfer of knowledge of criteria and methods of evaluating familiar designs for the task of evaluating a candidate solution to a new problem. In addition, for any of these design tasks, the transfer can take the form of strategic knowledge instead of domain knowledge. The transfer of a method for problem decomposition is but one example of this kind of analogy. The transfer of a design strategy in the form of a task structure is a more general example.

For all these reasons, we can generalize the traditional characterization of analogical design into an alternative characterization like this: analogical design involves

recall of knowledge about one design situation and transfer of that knowledge to another design situation, where the transfer can occur in the service of any design task in the new situation, and the content of the transfer can be knowledge of a design problem, solution, pattern, or strategy.

How? This question concerns methods for recall and transfer. The answer in general depends on the answers to the *why* and *what* questions about the design situation. The method for analogical transfer in design, for example, might depend both on the design task and on the content of the knowledge transferred.

Case-based reasoning (Kolodner, 1993) provides one general answer to the *how* question. Given a problem P_{new}, the designer is first reminded of a familiar problem P_{old} with a solution S_{old}, where P_{old} and P_{new} are so similar that S_{old} is an approximate solution to P_{new}. Then, the designer modifies selected components of S_{old} to obtain a candidate solution S_{new} to P_{new}. Thus, in case-based design, the entire solution S_{old} is transferred to S_{new} and modified to fit the specifications of P_{new}. Case-based reasoning, therefore, appears to be a limiting case of analogical reasoning in which the question of what to transfer degenerates into the question of what to modify. The case-based strategy, nevertheless, fits many tasks in variant and adaptive design.

The literature on case-based reasoning (Leake, 1996) distinguishes between transformational analogy (Winston, 1979) and derivational analogy (Carbonell, 1986). In transformational case-based reasoning, the solution S_{old} to P_{old} is modified to obtain the solution S_{new} to P_{new}. The knowledge about the modification is typically associative, and is often based on domain-specific heuristics. In derivational case-based reasoning, the trace of the problem-space search that led to the solution S_{old} to P_{old} guides the adaptation of S_{old}. Although both methods have been tried in design, they have met with only limited success. Associative methods for design modification appear adequate at best only for variant design in weakly interacting domains, and processing traces of the design generation by search are typically not available in practical design. Instead, research into the use of AI in design appears to have led to a third framework for case-based design that uses model-based methods for design modification and other subtasks spawned by the case-based strategy. These model-based methods use "deeper" design knowledge – for example, knowledge of the topology and teleology of S_{old}.

The literature on analogical reasoning (K. Holyoak, Gentner, & Kokinov, 2001) distinguishes between within-domain and cross-domain analogies. A domain can be characterized by the objects, relations, and processes that occur in it. Design domains such as engineering, architecture, software, and interface design clearly involve different kinds of objects, relations, and processes. But, from the viewpoint of building AI theories, whether two design domains are very different, quite similar, or identical depends on the language for representing the objects, relations, and processes. From this viewpoint, the design domains of mechanical and electrical engineering are very different if one uses different representational languages for them, or similar if one uses the same language. Whether an AI theory of design can use the same representational language for two (apparently different) domains depends on the level of design detail.

Again, the answer to the *how* question provided by case-based reasoning seems limited to design situations in which P_{old} and P_{new} are so similar that all of S_{old} can be transferred to S_{new} and selectively modified to fit P_{new}. What happens when P_{old} and P_{new} are not quite so similar? For example, what happens if P_{old} and P_{new} are problems from two different design domains in that not all the objects, relations, and processes in the two domains are identical? This question relates directly to the issue of creativity in design: if P_{old} and P_{new} are almost identical problems, S_{old} is unlikely to suggest changes to the variables characterizing P_{new}.

In general, analogical transfer requires the use of generic abstractions, where these abstractions typically express the structure of relationships between generic types of objects and processes. The literature on analogical reasoning suggests that these generic abstractions are not merely abstractions over features of objects, but that they capture the relational structure among objects and processes, for example (Gick & Holyoak, 1983). In the context of design, generic abstractions might specify, for example, the structure of geometric, topological, temporal, causal, and functional relations among design elements. That is, the abstractions may specify design patterns (Alexander, 1964). Generic design abstractions might also specify the structure of goals and methods in a design strategy. This implies that the learning of generic design abstractions is another important process in analogical design.

For this reason, we can specialize the initial characterization of analogy in design like this: analogical design involves learning of generic design abstractions from one design situation and their transfer to another, where these generic design abstractions specify the structure of relations among the elements of a design problem, solution, domain, or strategy, and where the transfer can occur in the service of any design task in the new situation.

When? This question pertains to the strategic control of processing. The learning of generic design abstractions provides a good illustration of this issue. Generic design abstractions can be learned at storage time, when the designer stores a new design in the design memory. This is an example of "eager" learning, in which abstractions over known designs are learned as soon as new design knowledge becomes available.

Alternatively, learning might occur at retrieval time, when the designer is reminded of a known design. As yet another alternative, learning might occur at problem-solving time, when the designer transfers knowledge from one design situation to another. The latter two are examples of "lazy" learning, in which abstractions are generated when needed. Clearly, different answers to the *when* question about this learning result in different computational theories of analogy-based creative design. In addition, the different answers to this question might imply different answers to the questions of what is abstracted and transferred and how.

7.4 Model-Based Analogy

In the mid–1990s, we developed a computational theory of analogy-based creative design called *model-based analogy* (Bhatta & Goel, 1996; Goel & Bhatta, 2004) to answer some of the above issues. We also developed the *Ideal* system, which instantiates and evaluates the theory for the conceptual design of engineering devices such as automatic coffee makers, electronic amplifiers, and gyroscopes. *Ideal* is a multistrategy system, capable of both case-based and analogical design. It contains several kinds of domain knowledge: design cases, design patterns, design concepts, generic design components, and generic domain substances. Design cases take the form of *structure–behavior–function* (SBF) models.

The SBF model of a device is based on a general ontology of the containment and flow of substances through device components (Goel, 2013b; Goel, Rugaber, & Vattam, 2009). This ontology gives rise to a set of design concepts in the form of structural connections and behavioral interactions among the components and substances of devices; behavioral states and state transitions in devices; and functions of devices and device components. These design concepts provide an SBF vocabulary for representing how devices work. The SBF model of a specific device explicitly represents the structural elements and topology of the device, the functions of the device and its components, and the internal causal processes (called *behaviors*) that explain the workings of the device. The design cases are indexed by their functions and organized around design concepts that specify the primitive functions of devices.

The design patterns in *Ideal* are generic abstractions of designs that the system has encountered in past design episodes. *Ideal* learns (and thus knows of) two kinds of design patterns: *general physical principles* (GPPs) and *generic teleological mechanisms* (GTMs). A GPP expresses a pattern of causal relations – for example, the causal relations that characterize the principle of heat flow from a hot body to a cold one. A GTM expresses a pattern of functional and causal relation – for example, the functional and causal relations that characterize feedback in control systems. Neither kind of design pattern in *Ideal* specifies information about the physical structure of devices. Both kinds are expressed in a BF (behavior–function) subset of the SBF language. Also, both kinds are indexed by the design goals they can help to accomplish.

Figure 7.1 illustrates a portion of *Ideal*'s computational process of model-based analogy. We will describe parts of the process that pertain to the learning, recall, and use of GTMs in analogical design in detail below. A design problem in *Ideal* specifies the functional requirements of the desired device along with behavioral and structural constraints (if any). For example, one design problem presented to *Ideal* specified the function of an operational amplifier as the desired function, with the additional requirement that the fluctuations in the output voltage be small. When a design problem is presented to *Ideal*, in the recall step it classifies the elaborated problem specification into its conceptually organized memory of design cases and retrieves the best-matching case.

This classification relies on the *functional similarity* between the design problem and the stored cases. In the example of the amplifier, if *Ideal* knows of an amplifier

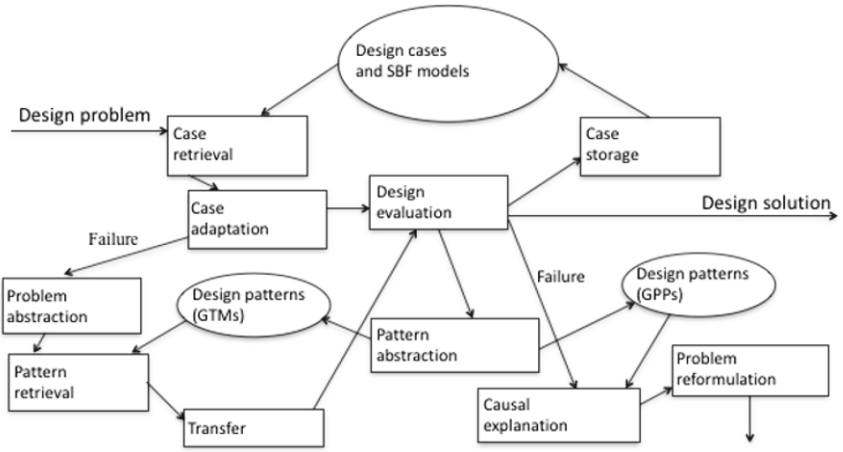


Fig. 7.1 *Ideal*'s computational process of model-based analogy. Model-based analogy is based on learning and transfer of design patterns such as generic teleological mechanisms (GTMs) and general physical principles (GPPs).

design that allows large voltage fluctuations, it retrieves this design because of its functional similarity to the desired design. Next, in the case adaptation phase, *Ideal* compares the function of the desired design and the function delivered by the retrieved design, and spawns adaptation goals in the form of differences between the desired function and the function delivered by the retrieved design. In the amplifier example, *Ideal* forms the adaptation goal of reducing the voltage fluctuations in the retrieved design.

The top left part of Fig. 7.2 indicates that *Ideal* retrieves the amplifier design from its case memory. Although not shown in Fig. 7.2, *Ideal* also retrieves the SBF model for the design. Briefly, a causal behavior is expressed as a directed graph of behavioral states and state transitions. This expresses each state as a schema that specifies a substance flowing be-tween components, its properties at a specific component, and the property values. The system also expresses each state transition as a schema that specifies the causes for the transition (for example, the function of a component) and relations between the property values in the preceding and succeeding states.

To understand the use of analogies in *Ideal*, consider what happens if the system does not initially know about control mechanisms such as feedback and feedforward. In this knowledge condition, the system searches the SBF model of the retrieved design but *fails* to localize the cause of large fluctuations in the output voltage.

Suppose that at this stage a teacher provides *Ideal* with a design for the desired amplifier along with its SBF model. The system stores the new design and its model in its analogue memory. But it also compares the SBF models of the old design (that allows large voltage fluctuations) and the new design (that regulates the voltage fluctuations). In particular, it inspects the structure of the causal behaviors in the SBF models of the two designs (top of Fig. 7.2), notes both the similarities and the differences, and abstracts a causal pattern corresponding to the difference between

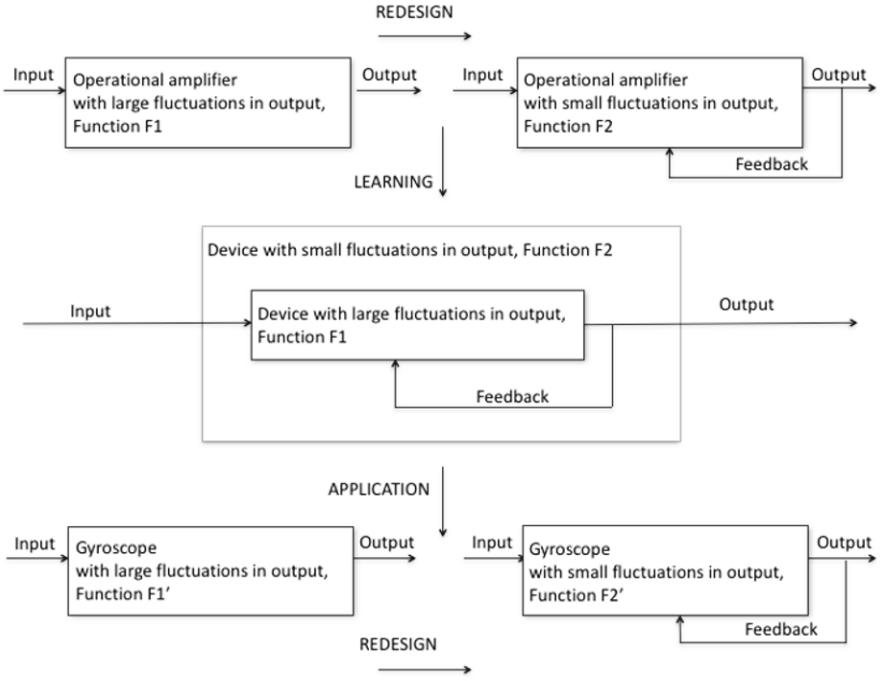


Fig. 7.2 An illustration of model-based learning and analogical transfer of a design pattern in *Ideal*. *Ideal* learns the generic teleological mechanism of closed loop feedback (shown at the center of the figure) from the domain of operational amplifiers (shown at the top) and transfers it to design a gyroscope with small fluctuations (shown at the bottom).

the two causal structures (middle of Fig. 7.2). In addition, it abstracts the adaptation goal (in the form of reducing a functional difference) that it had earlier failed to achieve, uses this functional abstraction as an index to the abstracted causal pattern, and stores the new functional and causal pattern in its memory of GTMs.

Figure 7.3 illustrates the GTM learned by *Ideal* from the amplifier example. In particular, the top part of Fig. 7.3 illustrates the specification of the generic function of reducing large fluctuations in the output of a device, and the bottom part illustrates the specification of the abstract causal pattern that reduces this kind of generic functional difference. The system expresses the generic function as a schema that specifies a desired function (F_2) and a known function (F_1), and relations between F_1 and F_2. It expresses each function in terms of the behavioral state it takes as input (*Given*) and gives as output (*Makes*), and expresses each state in terms of a substance (?*Sub*), its properties (for example, ?*prop*1), and their values (for example, *val*21 and *val*22). The Relationship slot (top right of the figure) specifies the relationship between F_1 and F_2. It specifies, for example, that *val*21 fluctuates by a small amount (little delta δ) whereas *val*22 fluctuates by a different and larger amount (big delta Δ). It also specifies that F_2 is related to F_1 through some hypothetical function f, where f is a map to an intermediate value (?*val*11′). The system expresses the abstract causal

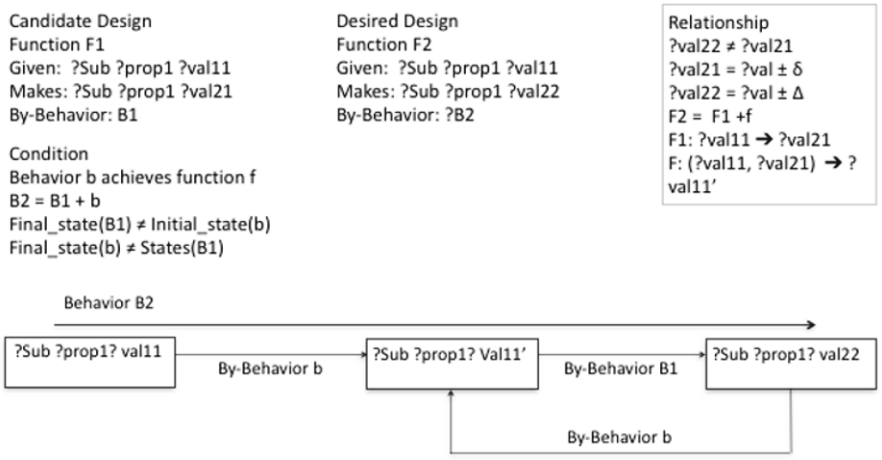


Fig. 7.3 *Ideal*'s specification of the learned design pattern.

pattern as a directed graph of behavioral states and state transitions that specifies that the behavior $B2$ for accomplishing F_2 can be achieved by a particular combination of the behavior $B1$ for accomplishing F_1 and a behavior b that accomplishes the function f. Behavior b is constrained by a specific relationship between the states in b and $B1$, and $B2$ combines $B1$ and b in a specific pattern of causal interaction

What happens when *Ideal* is given a new problem in a different domain? For example, one design problem actually given to the system at this stage specified the function of a gyroscope with the requirement that the fluctuations in the output angular momentum be below a specified limit, as illustrated at the bottom of Fig. 7.2. As discussed earlier, the system first classifies it into the primitive device functions, and then retrieves the best matching case. In the gyroscope example, it retrieves the design of a gyroscope that allows large fluctuations in the output angular momentum. As discussed earlier, the system compares the functions of the desired and the retrieved designs, and spawns adaptation goals. Again, it searches the SBF model of the retrieved design of the gyroscope, and again it fails to localize the cause for the functional difference. But then *Ideal* abstracts the adaptation goal and uses this abstraction as a probe into its memory of GTMs. It accesses the GTM it had learned in the previous design episode, illustrated in Fig. 7.3. Next, it instantiates the retrieved GTM in the context of the retrieved gyroscope design to generate an SBF model of a candidate design for the desired gyroscope.

At this processing stage, the candidate solution is only an initial conceptual design. *Ideal* further refines, evaluates, and completes the conceptual design. For example, it evaluates the candidate design for the gyroscope by tracing through the causal behaviors in the SBF model of the design, ensuring that the behaviors are internally consistent and lead to the achievement of the function desired of it. In this way, *Ideal* succeeds in generating a candidate design for the new design problem through analogical transfer of design patterns acquired from earlier design episodes.

This short description of *Ideal* does not cover many aspects of its processing. For example, *Ideal* also uses design patterns for assimilating information about a design failure and reinterpreting the design problem. In addition, it refines the abstracted design pat-terns in subsequent design episodes. *Ideal* uses generic design patterns for cross-domain analogical transfer, which introduces new variables into a design problem space – for example, control variables in the design of the gyroscope.

7.5 Biologically Inspired Design

In the mid–2000s, we started empirically studying biologically inspired design because it engages analogical thinking. Biologically inspired design (also known as biomimicry and biomimetics) is a growing movement in modern design that espouses the use of nature as an analogue for designing technological systems and processes (Benyus, 1997; Vincent, Bogatyreva, Bogatyrev, Bowyer, & Pahl, 2006; Vincent & Mann, 2002). This paradigm has inspired many designers in the history of design, such as Leonardo da Vinci and the Wright brothers. However, it is only over the last generation that the paradigm has become a movement, pulled in part by the growing need for environmentally sustainable development and pushed partly by the desire for creativity and innovation in design. The design of wind turbine blades mimicking the design of tubercles on the pectoral flippers of humpback whales is one example of biologically inspired design (AskNature, 2017). As Fig. 7.4 illustrates, tubercles are large bumps on the leading edges of humpback whale flippers that create even, fast-moving channels of water flowing over them. The whales can thus move through the water at sharper angles and turn tighter corners than if their flippers were smooth. When applied to wind turbine blades, they improve lift and reduce drag, improving the energy efficiency of the turbine. Figure 7.5 illustrates the design of the nose of the Shinkansen bullet train inspired by the design of the Kingfisher's beak: this design allows the train to travel at higher speeds with lower levels of noise (AskNature, 2011). Note that, by definition, the conceptual phase of biologically inspired design entails analogical transfer from biology to the design domain. Note also that the designs in Figs. 7.4 and 7.5 are novel, useful, feasible, and nonobvious, even surprising, and thus creative.

To make these abstract points tangible, let us briefly consider a couple of specific scenarios. Imagine an architect designing a high-rise building. The architect may need to find a mechanism for lifting water from the bottom of the building to its top. She might use current designs of electromechanical systems that can pump water hundreds of feet of high. However, these systems consume large amounts of energy. Now, one possibility is to monitor, model, analyze and optimize these water-pumping systems so that they work and are used more efficiently. This kind of design optimization sometimes can result in significant savings in critical resources such as energy and water, and thus we should pursue it. Another possibility, however, is to think about this problem in terms of the efficient – and thus, in the long run, more sustainable – mechanism of transpiration that redwood trees use to lift water

Fig. 7.4 Design of wind turbine blades to increase efficiency, inspired by the tubercles on humpback whale flippers.

Fig. 7.5 Design of Shinkansen bullet train's nose to decrease noise, inspired by the Kingfisher's beak.

hundreds of feet high. Of course this would require the invention of new materials that can support transpiration on the scale of a high-rise building. But this is part of the point: biologically inspired design encourages designers to view traditional problems from new perspectives.

Now consider a second and bigger design problem. Water is a scarce resource in many parts of the world. Desalination of ocean water offers an obvious solution to the problem of water scarcity. However, current technologies for water desalination are inefficient and costly. Yet, if we search for "water desalination" on, say, Google, then although we get a few million hits, all the millions of hits appear to refer to current technologies. This is a missed opportunity because there are a large number of biological organisms that perform water desalination quite efficiently, for example, some kinds of desert plants, snails, and mice. Nature provides the world's largest library of sustainable designs. So why not build a new generation of search engines that enable access to nature's design library? Why not reuse patterns of designs that

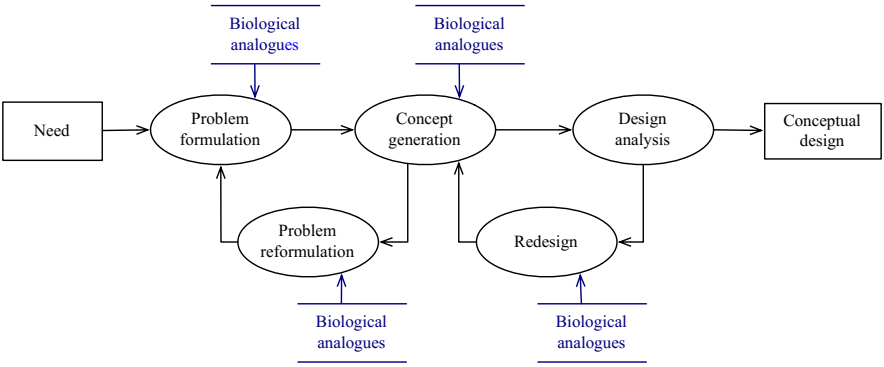


Fig. 7.6 A data flow diagram for a simplified version of the general process of preliminary design: ovals depict functions, rectangles depict inputs and outputs, and parallel horizontal lines denote data sources. The process starts with a need and results in one or more conceptual designs. It consists of the functions of problem formulation, concept generation, and design analysis. Generation of a design concept may lead to problem reformulation, and design analysis may lead to redesign. As indicated by the data sources in blue, biological analogies are useful for several functions in biologically inspired design, including concept generation, design analysis, redesign, and problem reformulation.

nature has already discovered over billions of years? As this example illustrates, biological cases may help designers spawn new problem spaces, which may lead to the invention of new technologies.

7.6 Model-Based Analogies in Biologically Inspired Design

Recently there has been considerable research on studying biologically inspired design from an information-processing perspective and on developing computational techniques to support its practice; see, for example, Goel, Vattam, Helms, and Wiltgen (2014); Goel, McAdams, and Stone (2014). Our in situ observations of biologically inspired design in practice (Yen, Helms, Goel, Tovey, & Weissburg, 2014) have indicated the use of multiple processes, including both problem-driven analogy and solution-based analogy (Helms, Vattam, & Goel, 2009), and entailing compound analogies (Vittam, Helms, & Goel, 2008). Figure 7.6 illustrates a simplified version of the general process of problem-driven analogy for generating conceptual designs. As illustrated in the figure, biological analogies are useful in several tasks of preliminary design, including concept generation, design analysis, redesign, and problem reformulation (Vittam, Helms, & Goel, 2010).

Figure 7.7 illustrates a simplified version of the general process of problem-driven concept generation in biologically inspired design in more detail (Goel, Vattam, et al., 2014). Fig. 7.7 also indicates some of the fundamental roles AI can play in systematizing the design process as well as biological knowledge from a design

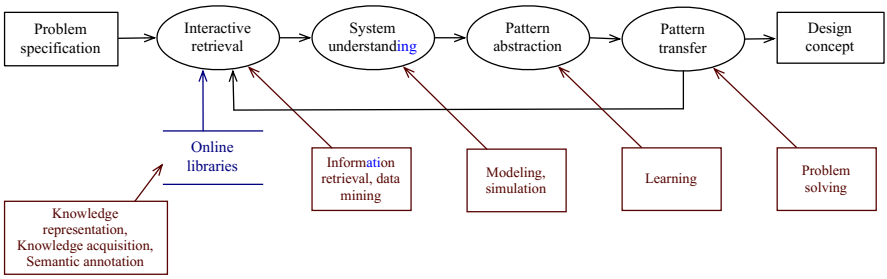


Fig. 7.7 A data flow diagram of the process of problem-driven analogical concept generation in biologically inspired design. The process consists of interactive retrieval of biological cases online, understanding the biological systems, abstracting design patterns, and transferring the patterns to the given design problem. The design process is iterative. The brown boxes indicate the fundamental roles that AI can play in systematizing the design process, as well as biological knowledge from a design perspective.

perspective. Let us analyze biologically inspired design in terms of the four questions we presented earlier.

Why? The *why* question refers to the task for which analogies are used. As Fig. 7.6 indicates, problem-driven biologically inspired design uses analogies for multiple design tasks, especially concept generation, design analysis, redesign, and problem reformulation. More generally, it seems that analogies are useful for almost any conceptual design task, including interpreting the design problem, decomposing the problem, generating a solution to a problem or subproblem, composing sub-problem solutions into a candidate design, anticipating potential difficulties with a candidate design, evaluating a candidate design, interpreting the evaluation information, and reformulating the design problem. *Ideal*'s technique of model-based analogy uses analogies for completing a partial solution to a design problem; in *Ideal*, the partial solution might be null. *Ideal* also uses analogies for interpreting the evaluation of a design. For what other design tasks might model-based analogy be especially useful? In more recent work, we have developed similar techniques of model-based analogy for several design tasks ranging from understanding design drawings (Yaner & Goel, 2008) to evaluating design concepts (Wiltgen & Goel, 2016).

What? The *what* question pertains to the knowledge that is transferred from a source analogue to a target problem. Biologically inspired design entails analogical transfer of several kinds of knowledge, including functions, causal mechanisms, problem decompositions, and design patterns (Vittam et al., 2010). In *Ideal*, analogical transfer is mediated by generic design abstractions in the form of design patterns. *Ideal* uses functional and causal patterns with no spatial information. In more recent work, we have developed similar techniques of model-based analogy for understanding design drawings that include spatial knowledge in the design patterns (Yaner & Goel, 2008).

How? The *how* question concerns the mechanisms by which knowledge is transferred from source cases to a target problem. As Fig. 7.7 indicates, biologically inspired design engages a variety of mechanisms of analogical transfer, including

accessing biological cases from external information sources such as the web, duplication of functions, abstraction, and transfer of design patterns. *Ideal* uses model-based methods for analogical transfer; *Ideal*'s models specify functional, behavioral, causal, and structural knowledge. In recent work, we have used model-based methods to access biological cases from biology articles on the web (Vattam & Goel, 2014).

When? The *when* question refers to the stage in analogical reasoning when abstractions are learned and used. *Ideal* learns design at storage time; *Ideal* is an example of an eager learner. In biologically inspired design, abstractions are learned at several different stages, including storage, retrieval, and transfer.

7.7 Conclusions

Research on computational creativity is maturing. A major part of the research agenda on computational creativity is to characterize creative tasks and identify the processes of creativity (Boden, 2009; Wiggins, 2006). In this chapter, we studied conceptual system design is a creative task and analogical thinking as a fundamental process of creativity.

Current theories of analogy entail retrieval of source cases from memory, learning of generic abstractions, and their transfer to a target problem. The *Ideal* system uses model-based analogy to perform analogical design. In particular, *Ideal* retrieves *structure–behavior–function* models of designs in its memory, learns design patterns such as *generic teleological mechanisms*, and transfers these mechanisms to the target problem.

Biologically inspired design provides a real-world task domain to investigate analogy. As the analysis in this chapter indicates, analogical thinking in biologically inspired design is much more varied than current theories of analogical reasoning. Biologically inspired design entails the use of multiple processes of analogical thinking such as problem-driven analogy and solution-based analogy, and engages compound analogies. Further, biological analogies can be used for several tasks of preliminary design, including concept generation, design analysis, redesign, and problem reformulation

Thus, analogical thinking in practice presents a challenge to the application of existing AI methods. On the other hand, it also provides opportunities for developing new theories, representations, methods, and architectures for analogical thinking.

Acknowledgements I started working on computational design, analogy, and creativity around 1985. I am grateful to my many collaborators over the last thirty years who have deeply influenced my research, including Sambasiva Bhatta, Balkrishnan Chandrasekaran, Michael Helms, Janet Kolodner, Sattiraju Prabhakar, Spencer Rugaber, Swaroop Vattam, Bryan Wiltgen, Patrick Yaner, and Jeannette Yen. I am also thankful to Amílcar Cardoso and Tony Veale for putting this volume together and inviting me to write this chapter, and to Pedro Martins and Gözde Özbal for their helpful critiques of an earlier draft.

This chapter is based in large part on Goel (1997) and Goel (2013a). Goel (1997) characterized creativity in design, described analogical thinking as a core process of design creativity, and proposed an agenda for research on analogical design. Goel (2013a) analyzed biologically inspired design from the perspectives of analogy and creativity. This chapter bridges and builds on the two articles: It revisits the framework for analyzing analogical reasoning, examines biologically inspired design as a task domain for studying analogical reasoning, and reflects on selected facets of research on computational design, analogy and creativity.

References

Alexander, C. (1964). *Notes on the Synthesis of Form*. Cambridge, MA: Harvard University Press.

Arnheim, R. (1969). *Visual Thinking*. Berkeley, CA: University of California Press.

AskNature. (2011). Shinkansen train. https://asknature.org/idea/shinkansen-train#.WhqZfLacaV4. Accessed: 2011-04-28.

AskNature. (2017). Flippers provide lift, reduce drag. https://asknature.org/strategy/flippers-provide-lift-reduce-drag#.WhqYeLacaV6. Accessed: 2017-11-11.

Benyus, J. (1997). *Biomimicry: Innovation Inspired by Nature*. New York: William Morrow.

Bhatta, S., & Goel, A. (1996). From design experiences to generic mechanisms: Model-based learning in analogical design. *AIEDAM*, *10*(1), 131–136.

Boden, M. (2009). Computer models of creativity. *AI Magazine*, *30*(3), 21–34.

Brown, D. (1996). Routineness revisited. In M. Waldron & K. Waldron (Eds.), *Mechanical Design: Theory and Methods* (pp. 195–208). Berlin: Springer.

Brown, D., & Chandrasekaran, B. (1989). *Design Problem Solving: Knowledge Structures and Control Strategies*. San Mateo, CA: Morgan Kaufmann.

Carbonell, J. (1986). Derivational analogy: A theory of reconstructive problem solving and expertise acquisition. In R. Michalski, J. Carbonell, & T. Mitchell (Eds.), *Machine Learning: An Artificial Intelligence Approach* (Vol. 2). Los Altos: Morgan Kaufmann.

Colton, S., & Wiggins, G. (2012). Computational creativity: The final frontier? In *Proceedings of ECAI-2012, 20th European Conference on Artificial Intelligence*, Montpellier, France.

Cox, M., & Raja, A. (2011). *Metareasoning: Thinking about Thinking*. Cambridge, MA: MIT Press.

Cross, N. (2011). *Design Thinking: Understanding How Designers Think and Work*. New York: Berg.

Dym, C., & Brown, D. (2012). *Engineering design: Representations and Reasoning* (2nd ed.). New York: Cambridge University Press.

Forrester, J. (1961). *Industrial Dynamic*. Cambridge, MA: MIT Press.

Gick, M., & Holyoak, L. (1983). Schema induction and analogical transfer. *Cognitive Psychology*, *15*(1), 1–38.

Goel, A. (1997). Design, analogy, and creativity. *IEEE Expert*, *12*(3), 62–70.

Goel, A. (2013a). Biologically inspired design: A new program for computational sustainability. *IEEE Intelligent Systems*, *28*(3), 80–64.

Goel, A. (2013b). One thirty year long case study; fifteen principles: Implications of an AI methodology for functional modeling. *AIEDAM*, *27*(3), 203–215.

Goel, A., & Bhatta, S. (2004). Use of design patterns in analogy-based design. *Advanced Engineering Informatics*, *18*(2), 85–94.

Goel, A., McAdams, D., & Stone, R. (Eds.). (2014). *Biologically Inspired Design: Computational Methods and Tools*. Berlin: Springer.

Goel, A., Rugaber, S., & Vattam, S. (2009). Structure, behavior, and function of complex systems: The structure, behavior, and function modeling language. *Artificial Intelligence for Engineering Design, Analysis and Manufacturing*, *23*(01), 23–35.

Goel, A., Vattam, S., Helms, M., & Wiltgen, B. (2014). Information-processing accounts of biologically inspired design. In A. Goel, D. McAdams, & R. Stone (Eds.), *Biologically Inspired Design: Computational Methods and Tools* (pp. 127–152). Berlin: Springer.

Helms, M., Vattam, S., & Goel, A. (2009). Biologically inspired design: Process and products. *Design Studies*, *30*(5), 606–622.

Holyoak, K., Gentner, D., & Kokinov, B. (Eds.). (2001). *The Analogical Mind: Perspectives from Cognitive Science*. Cambridge, MA: MIT Press.

Holyoak, K., & Thagard, P. (1996). *Mental Leaps: Analogy in Creative Thought*. Cambridge, MA: MIT Press.

Josephson, J., & Josephson, S. (1996). *Abductive Inference: Computation, Philosophy, Technology*. New York: Cambridge University Press.

Kolodner, J. (1993). *Case-Based Reasoning*. San Mateo, CA: Morgan Kaufmann.

Leake, D. (Ed.). (1996). *Case-Based Reasoning: Experiences, Lessons, and Future Directions*. Cambridge, MA: MIT Press.

Newell, A., & Simon, H. (1972). *Human Problem Solving*. New York: Prentice-Hall.

Sternberg, R. (Ed.). (1999). *Cambridge Handbook of Creativity*. New York: Cambridge University Press.

Thagard, P. (2005). *Mind: Introduction to Cognitive Science*. Cambridge, MA: MIT Press.

Vattam, S., & Goel, A. (2014). Computational model of interactive analogical retrieval. In H. Prade & G. Richard (Eds.), *Computational Approaches to Analogical Reasoning: Current Trends*. Berlin: Springer.

Vincent, J., Bogatyreva, O., Bogatyrev, N., Bowyer, A., & Pahl, A. (2006). Biomimetics: Its practice and theory. *Journal of the Royal Society Interface*, *3*, 471–482.

Vincent, J., & Mann, D. (2002). Systematic transfer from biology to engineering. *Philosophical Transactions of the Royal Society of London*, *360*, 159–173.

Vittam, S., Helms, M., & Goel, A. (2008). Compound analogical design: Interaction between problem decomposition and analogical transfer in biologically inspired design. In *Proceedings of the 3rd International Conference on Design Computing and Cognition* (pp. 377–396). Atlanta: Berlin: Springer.

Vittam, S., Helms, M., & Goel, A. (2010). A content account of creative analogies in biologically inspired design. *AIEDAM*, *24*, 467–481.

Wiggins, G. (2006). A preliminary framework for description, analysis and comparison of creative systems. *Knowledge-Based Systems*, *19*(1), 449–458.

Wiltgen, B., & Goel, A. (2016). Functional model simulation for evaluating design concepts. *Advances in Cognitive Systems*, *4*, 151–168.

Winston, P. (1979). Learning and reasoning by analogy. *Communications of the ACM*, *23*(12), 1–38.

Yaner, P., & Goel, A. (2008). Analogical recognition of shape and structure in design drawings. *AI for Engineering Design, Analysis and Manufacturing*, *22*(2), 117–128.

Yen, J., Helms, M., Goel, A., Tovey, C., & Weissburg, M. (2014). Adaptive evolution of teaching practices in biologically inspired design. In A. Goel, D. McAdams, & R. Stone (Eds.), *Biologically Inspired Design: Computational Methods and Tools* (pp. 153–200). Berlin: Springer.

Chapter 8
The Evaluation of Creative Systems

Graeme Ritchie

Abstract As creative systems become more advanced, there is a growing need to assess their performance, whether during development or after implementation is complete. There is not yet any settled methodology for evaluation, despite a continuing debate. The task requires answers to two methodological questions: which properties of the behaviour of a creative computational system should be considered, and what are suitable ways of measuring these properties? It is essential to take into account the long-term aim of the work being evaluated, as different theoretical agendas may lead to variations in evaluation requirements. We review some of the theoretical and methodological suggestions that have been made, ranging from candidates for essential ingredients of creativity to practical matters about rating the output of programs. Many of the essential judgements about the success of a creative system can be made only subjectively, which means that it can be useful to borrow methods from experimental psychology. Nevertheless, evaluation within computational creativity has its own unique attributes, requiring specific approaches.

8.1 The Need for Evaluation

When software is built, whatever the motivation, a natural question to ask is: "how well does this program perform?" The exact notion of "performing well" will vary greatly between different types of software, and it may not always be obvious how best to define this. Nevertheless, the question is important. This is what we will refer to here as *evaluation*: a considered and systematic assessment of how successful a piece of software has been, where "successful" is broadly interpreted and depends very much on the aims of the software's designer.

Graeme Ritchie
Department of Computer Science, University of Aberdeen, Aberdeen AB24 3UE, United Kingdom
e-mail: g.ritchie@abdn.ac.uk

T. Veale, F. A. Cardoso (eds.), *Computational Creativity*, Computational Synthesis and Creative Systems, https://doi.org/10.1007/978-3-319-43610-4_8

In the early days of artificial intelligence (AI), evaluation of supposedly intelligent systems was more or less non-existent. Even major historic programs (e.g., (Lenat, 1976; Winograd, 1972; Winston, 1970)), were reported anecdotally, primarily using samples of the program's output, which might have been picked to be particularly impressive. Rigorous controlled testing and thorough scientific reporting of outcomes were not expected of such "exploratory" programs. (Pithy critiques of early AI practices were presented by Hayes (1975) – reprinted in (Hayes & Ford, 1999) – and by McDermott (1976) – reprinted in some editions of (Haugeland, 1981).) However, for the past few decades, AI has increasingly incorporated systematic evaluation of techniques and of programs into its normal methodology. A landmark in this development was Cohen (1995), which showed in some detail how empirical methods developed within the social sciences could be applied to the testing of AI programs such as planners.

Computational creativity (CC), as a slightly younger subfield, lags slightly behind mainstream AI, but is catching up rapidly. There has been much explicit debate about this issue within CC, both with regard to specific domains (Linson, Dobbyn, & Laney, 2012; Norton, Heath, & Ventura, 2015; Zhu, 2012) and also regarding CC in general (Colton, Pease, Corneli, Cook, & Llano, 2014; Jordanous, 2011, 2014; Lamb, Brown, & Clarke, 2015; Pease, Winterstein, & Colton, 2001). Despite this, there is not yet a consensus on how best to evaluate CC systems, and thorough evaluation is still not a routine part of CC research. Pearce, Meredith, and Wiggins (2002, p. 119) lamented this "malaise" in the area of the automated composition of music, and the problem has also been noted for story-generation:

> Because the issue of what should be valued in a story is unclear, research implementations tend to sidestep it, generally omitting systematic evaluation in favor of the presentation of hand-picked star examples of system output as means of system validation.
>
> Gervás (2009, p. 61)

Our examination of evaluation is framed in the context of a scientific or engineering perspective on CC, as opposed to an artistic stance. Outside AI, various creative practitioners have made direct use of technology to construct novel aesthetic artefacts, such as music or visual art. Although it may be interesting to assess such artistic work in various ways (e.g. how it fits into a cultural context), that is not the type of evaluation which we shall be focusing on in this chapter.

If the goal of the activity is to gain a clearer scientific understanding either of a particular domain (music, narrative, visual art, etc.) or of creativity in general, then evaluation becomes crucial, since we have to know how different abstract models operate, and which theories better characterise the creative activity. Similarly, if the aims are viewed as technological, to develop practical tools for creating artefacts for real audiences (for example, to write novels for publication), then it is again important to discover which techniques work better, in some sense of "better" which has to be determined. We shall therefore assume here that the activity under consideration is the development and refinement of some abstract model of a particular domain, and that the overall goal is either science (gaining a better understanding, at a theoretical level, either of that domain or of creativity), or engineering (achieving better output

as a way of supplying worthwhile artefacts). It is beyond the scope of this chapter to examine wholly artistic projects which happen to use computational methods.

Although creativity is central to the CC field, this does not mean that there is a clear or widely accepted definition of this abstract concept. Any notion of "creativity" adopted in CC is derived from observations about creativity in humans, but there is not an agreed view of what is meant by human creativity. It has been argued (Colton, Cook, Hepworth, & Pease, 2014; Jordanous, 2012b) that creativity (both human and computational) may be what is known in philosophy as an *essentially contested concept* (Gallie, 1956). Gallie proposes some criteria for a concept to have this status, of which the most obviously relevant here are that it must signify a "valued achievement" which is internally complex and whose value relies on the component features of the achievement, and, moreover, people vary in their notions of the correct use of the concept, while being aware, at least to some extent, of how others use it.

The debate about how best to evaluate CC systems could be viewed (depending on the exact objectives of the research – see Section 8.2.3) as an operational version of the more abstract question "what is creativity?". In view of the profound lack of any accepted definition, we shall not attempt to solve the problem of characterising what constitutes creativity in the abstract, although we shall try to clarify particular uses of the term "creative" (Sections 8.2.1 and 8.2.3), and also review some ideas about creativity which have had an impact on discussions of evaluation (Section 8.3). Our main focus will be on reviewing some of the issues which arise in the evaluation of potentially creative computational systems, drawing attention to certain conceptual issues and methodological choices along the way.

Despite the implicit influence of opinions about creativity in humans, we shall not be reviewing evaluations of human creativity (Puccio & Murdock, 1999). That is, human creativity is an unavoidable presence in any discussion of CC, but it is not the subject of this chapter, and its evaluation is an entirely separate topic.

8.2 The Nature of the Task

8.2.1 Two Meanings of "Creative"

The terms "creative" and "creativity" have been in use for a long time within ordinary, non-scientific discourse, where there is no need for these words to be defined very exactly. Since science and engineering require some degree of rigour and precision, it is appropriate to look more closely at the usage which has grown up around CC research.

The first meaning of "creative" that is pertinent here is what we will call the *loose* interpretation of the term. This sense refers to activities which are treated, within wider society, as being "creative" by their very nature: fine art, writing poetry or fiction, composing music, etc. This sense is applied to activities, not to achievements,

and has little or no implication of excellence or innovation. If an ordinary person has a hobby of watercolour painting, they may well be congratulated on pursuing a creative activity, regardless of their level of expertise. On the other hand, much of society has not yet accepted that computer programming is a "creative" hobby, even if pursued with some degree of skill. Thus there is a general, vague categorisation of certain culturally defined activities as being "creative" and others as not being so. This sense is still widely used within CC research, since academic forums which solicit and accept contributions in the area of "creative systems" appear to apply this loose sense as a sufficient criterion for topical relevance. If a program generates poetry or music, it counts as a "creative system" in this sense, regardless of how bad or unoriginal the output. Thus the term is widened from the activity itself to a system which carries out that activity.

The second meaning of "creative", the *strict* usage, attributes a degree of excellence to the agent, and/or the activity, and/or its outcome. This is the sense in which one artist may be said to be more creative than another, or in which some achievement may be described as "truly creative". In CC, being strictly creative is a goal to be pursued, and the exact nature of what constitutes creativity in this strict sense is a matter of considerable debate, very relevant to the question of evaluation. Moreover, this notion of creativity is sometimes attributed to computer systems which are not creative in the loose sense, in that they perform in a domain (e.g. mathematics) which is not normally classed as creative by the lay person. (Ritchie (2011) used the terms "weak" and "strong" for the loose/strict distinction, but these terms have now been assigned a different meaning within the field – see Section 8.2.3.)

This ambiguity of usage, if not fully recognised, can lead to confusion:

> . . . computational systems are being presented as "creative systems" without their creativity being justified; hence "creative" becomes a descriptor of a system. . . . Using this key objective as a descriptor of the outcome, without appropriate justification, is not a suitable way of demonstrating that the objective has been met.
>
> Jordanous (2012a, p. 252)

In our tour of evaluation within CC, the type of computer system being considered will largely be creative systems in the loose sense (creative$_L$). However, being strictly creative (creative$_S$) will be considered as a possible success criterion for such systems. In other words, we will often be posing the question: how could we determine whether a creative$_L$ system is being creative$_S$? (This shows that the two senses can occur in a single meaningful sentence.)

The "loose"/"strict" distinction should not be confused with the "little-c"/"Big-C" notions of creativity. Kaufman and Beghetto (2009, p. 1) give examples of "everyday creativity. . . in which the average person may participate each day. . . e.g. creatively arranging family photos in a scrapbook; combining left over Italian and Chinese food to make a tasty, new fusion of the two cuisines. . . ". Such achievements exemplify *little-c* creativity. In contrast, studies of "eminent creativity" aim to learn about "creative genius" and "which creative works may last forever". This is *Big-C* creativity. In our terms, "little-c" and "Big-C" are two levels of "strict" creativity; that is, the "little-c"/"Big-C" distinction is about the extent to which something is creative$_S$.

8.2.2 Characteristics of a Creative$_L$ System

We have said that the computer systems under consideration here are, broadly, those which are creative$_L$. It is hard to be definitive about the typical computational characteristics of such systems, but the following attributes (Ritchie, 2007, p.71) capture the general picture (this is very much a rough prototypical description, as not all CC systems conform wholly to these criteria):

- The program generates members of some class of artefacts, which are usually culturally defined, and can be abstract rather than concrete. The class of artefacts exists prior to the program being constructed, and is not defined in terms of the workings of the program.
- The set of acceptable artefacts (i.e. successful outputs) is not well defined.
- The set of acceptable artefacts is very large (possibly infinite).
- Deciding on the acceptability of an artefact depends on factors which are social, cultural, subjective, and personal.
- In addition to mere acceptability, artefacts can vary in a further attribute of *quality* (sometimes known as *value*), judgement of which is also social, cultural, subjective, personal.

We will return to some of these properties, particularly the notions of acceptability and quality, in later sections.

Wiggins has presented an abstract characterisation of the way that a creative system (creative$_L$ in our terms) operates (G. Wiggins, 2001, 2003, 2005, 2006a, 2006b, 2017), sometimes referred to as the *creative systems framework* (CSF). This depicts the system as searching through a space of possible concepts, guided by three kinds of rules: those which define that space (which more or less corresponds to "acceptability"), those which rate the concepts (corresponding to "quality"), and those which indicate the trajectory through that space.

To sharpen up the distinction between a creative$_L$ system and a more conventional software, non-CC application, we can highlight some typical aspects of a creative$_L$ system which differentiate it from a more conventional non-creative$_L$ computer program:

- There is no practical task that is being achieved or facilitated.
- There is no precise definition of correct behaviour or error cases.
- There is no prior specification of precise or formal requirements.
- There is usually no objective (automatable) measure of success (but see Section 8.6.6).
- When considering the internal workings of a program, the criteria for what counts as a "good" method of implementation differ in some ways from traditional software guidelines.

Despite these observations, many of the traditional precepts of software engineering still apply. In building a creative$_L$ system, it is still important to have a modular

design, to write clean, perspicuous code, to debug systematically, to maintain versions, to document, and so on. The basic crafts of software are applicable, it is just that the bigger picture – what the program is supposed to be doing – is different.

By way of an example, compare what would be involved in the design and implementation of a graphical editing tool such as Adobe Photoshop or The Gimp, which provide graphical editing facilities to the user, and the development of The Painting Fool (Colton, 2012; Colton & Pérez-Ferrer, 2012), which creates its own digital paintings. In all these cases, general guidelines for hygienic programming (should) apply, but for a program like The Painting Fool further questions arise, such as the extent of the autonomy of the program and the quality and novelty of its outputs.

8.2.3 Varieties of Goals

The terms "creative$_L$" and "creative$_S$" as we defined them above are properties that can be attributed to systems, roughly glossed as "involving artistic activity" and "achieving genuine creativity" respectively. A related but slightly different set of distinctions can be made between the kinds of *aims* that research in CC may have:

> Often the aim of evaluation has been to see whether the systems contribute high-quality results to a creative domain, for example if the results are aesthetically pleasing, highly valuable, accurate or if they compare favourably to a test set of typical results, or if the processes used by the system are of particular interest. This is related to but distinct from the aim of whether the systems can demonstrate behaviour that can be seen as creative . . . These evaluative aims of quality and creativity can be confused, especially in the absence of a standard evaluation methodology for creativity, though these aims should not be treated as being mutually exclusive.
>
> Jordanous (2012a, p. 252)

That is, some CC work is primarily aimed at building software which is capable of generating artefacts, in some specific domain, which are as good as possible, within the constraints and norms of that (usually artistic) domain. (We shall return in later sections to the notion of what is meant by "good".) This will involve building a creative$_L$ system, but the question of whether the constructed system is creative$_S$ is not central. Colton, Pease, Corneli, Cook, and Llano (2014) use the term *weak* to refer to CC objectives of this sort. In contrast, they use the term *strong* for a CC objective which involves investigating the nature of (computational) creativity, in which any constructed system would be judged principally on whether it was creative$_S$. A researcher following "strong" CC objectives is not content with a program which can build well-formed and pleasing exemplars within an artistic (creative$_L$) domain: the real problem is beyond that, asking whether the artefact-building has been creative$_S$. The aim is to use the domain model only as a stepping-stone to a more abstract result, a (computational) model of creativity. This has consequences for how a CC system is evaluated, as will be explored in later sections.

There is yet another important conceptual distinction to be made, between two subtypes of Colton et al.'s "weak" objectives; this is roughly between an engineering motivation and a scientific motivation. The engineering approach wishes to build a creative$_L$ system for some particular purpose. For example, the goal might be to write simple jokes for children, or to design desktop wallpaper images for computers. Here, the aim is not creativity$_S$ itself, but the system builders will certainly want the output to be well-formed, as "good" as possible, and preferably novel.

The scientific outlook, on the other hand, typically explores a particular domain (e.g. visual art, narrative) in order to gain a better understand of that domain. The aim is to investigate the concepts which underlie artefacts in that domain, and how these concepts relate to each other. These goals might be shared with researchers in other disciplines, such as linguistics or psychology. Because CC, by definition, involves computation, such research also has the aim of determining how the underlying abstractions may be operated upon algorithmically to create artefacts. As with the engineering approach, there is some interest in ensuring that generated artefacts are of a good standard, but these success criteria are invoked as a check on the accuracy of the domain model, not for their practical use (as in weak engineering CC) or as potential indicators of strict creativity (as in strong CC). A clear example of this type of project is an early thesis on computer-generated jokes (Binsted, 1996), which contains no discussion at all of creativity, because that project was narrowly focused on modelling the structure of a genre of text, with the computer implementation allowing the consequences of the model to be tested.

It is less clear that work with strong CC objectives displays this distinction between scientific and engineering goals. Projects focusing centrally on strict creativity appear to be espousing scientific aims, in that their priority is to explore the nature of the phenomenon of (computational) creativity, rather than to build working artefacts for practical purposes.

The contrast between weak and strong objectives can be illustrated by relatively recent work in CC. Harmon (2015) describes a system which generates "creative" figures of speech, but the design and assessment of the system are very much focused on the production of "good" metaphors (i.e. modelling the chosen phenomenon), rather than modelling (strict) creativity as such. Tobing and Manurung (2015)'s description of their poetry generator is entirely concerned with the crafting of a poem, rather than testing the system's strict creativity. Hence, these two projects appear to have "weak" CC objectives. On the other hand, Grace and Maher (2015) explore how notions such as unexpectedness and surprise can be incorporated into a "transformational" model of creative activity (Boden, 1992), and Takala (2015) uses neural networks to model various creative effects directly, with the illustrative domain being only incidental; this seems to exemplify "strong" CC objectives.

Generally, the divergence between weak and strong objectives, and between the engineering and scientific motivations, leads to differences in emphasis concerning what should be evaluated, but there are some large overlaps in what factors are considered.

The term *mere generation* is sometimes used to describe certain work in CC (Cook, 2015, p.201), often in a pejorative way (Veale, 2014, p. 245), (Corneli et al.,

2015, p. 270). Although there is no clear definition of what is meant by this term (see (Ventura, 2016) for an interesting discussion), it sometimes appears to refer to work carried out with "weak" CC objectives. That is, there seems to be a view that such work (i.e. not primarily tackling the issue of *creativity*$_S$) is of a less interesting nature. However, this is entirely a personal judgement about which goals should be pursued, and in no way undermines the validity of "weak" CC research. Whether such work falls within the boundaries of CC is a separate question – Colton and Wiggins (2012, p.25) appear to suggest that it should not. To judge by the proceedings of the annual International Conference on Computational Creativity over the past few years, there is no shortage of projects with weak CC objectives.

8.2.4 Two Kinds of "Evaluation"

Another important distinction is whether the "evaluation" is a component of the computational model or is a methodological tool for assessing the success of the system.

The idea of "evaluation" as an internal component (of a creative system) recurs in the literature. It has been argued that, for a system to be creative$_S$, it must have an "appreciation" of the value of its own creations (Colton, 2008; Jennings, 2008), and Eigenfeldt, Bown, Brown, and Gifford (2016) quote Veale (2015) as arguing that the term "mere generation" (Section 8.2.3) describes systems which lack any reflection on their own creations. In Wiggins' abstract description of the workings of a creative$_L$ system (see Section 8.2.2), there is a formal evaluation measure, $\mathscr{E}$, which is used to assess the "value" or "quality" of generated items.

It is possible to imagine a very simple "generate-and-test" architecture for a creative$_L$ system in which there is an initial relatively unintelligent stage which proposes a large set of candidates to a later evaluative stage, which filters out any items of low quality. Although such an arrangement might be seen as an over-simplification, a subtler version can be found in the basic architecture of an evolutionary algorithm, where there is repeated use of a "fitness function" to rate candidate solutions. Within CC, there are other designs which contain some mechanism for the system to rate its own generated structures (Macedo & Cardoso, 2001; Norton, Heath, & Ventura, 2010). In all these cases, the "evaluation" is part of the model with which the system operates, guiding the computations and having an impact on which artefacts (or behaviours) are eventually displayed to the outside world. We will call this *internal* evaluation.

In contrast to this, there is the need (Section 8.1) for the research community to have an indication of how well a creative$_L$ system has performed. Such a system is designed in the hope that it will achieve some particular output or behaviour which meets some pre-existing ideas, either about artefacts in a particular domain, or about strict creativity. These ideas originate outside the system itself, for example from a theory about creativity or from social conventions, and they determine what counts

as success for the system as a whole. Testing the system's output/behaviour against such objectives is what we will call *external evaluation*.

The term "evaluation" is used in both these senses within CC. For example, Maher, Brady, and Fisher (2013) and Grace and Maher (2014) discuss possible internal evaluation methods, whereas Gervás (2002) and Jordanous (2012a) are concerned with external evaluation. (This difference between internal and external evaluation should not be confused with the distinction sometimes drawn between intrinsic and extrinsic motivation (Bénabou & Tirole, 2003; Ryan & Deci, 2000), despite at least one proposal to treat internal evaluation as modelling intrinsic motivation (Schmidhuber, 2010).)

This article is concerned with *external* evaluation (but see Section 8.6.6).

8.3 Theoretical Concepts

8.3.1 Descriptions, Causes and Symptoms

It would be logically possible to treat creativity as a monolithic, indivisible, concept, and for evaluation of strict creativity to ask just one question: has this system behaved creatively? Although this would amass a corpus of judgements which could be analysed in various ways, it would not in itself throw light on what factors are involved in (computational) creativity. Instead, various attempts have been made to dissect the notion of creativity, so as to identify separate features which, in suitable combinations, result in creativity. Such features are generally presented as facets of some overall framework, either for evaluation or for CC more generally.

Within CC, such frameworks have taken different perspectives. The *descriptive* position sets out to describe, as formally and precisely as possible, what takes place within creative$_L$ activity. Although such an account provides a framework within which hypotheses about the nature of creativity$_S$ can be stated, it does not inherently make predictions about ways in which successful (strict) creativity occurs. This is typified by Wiggins' CSF (see Section 8.2.2) and the *FACE* model (Colton, Charnley, & Pease, 2011; Pease & Colton, 2011a). These frameworks describe, in structural and/or procedural terms, the abstract architecture of a creative$_L$ system, without indicating which interactions of these components are creative$_S$, and without regard for which of these aspects are directly observable.

In contrast, a *causal* analysis tries to identify the preconditions or actions that, when they occur together, result in or constitute a creative$_S$ event; see Sections 8.3.2 and 8.3.4 below. It may employ the constructs of a descriptive framework in order to state these conjectures. The aim is to specify exactly what eventualities, at some abstract level, give rise to creative$_S$ behaviour.

Yet a third viewpoint could be described as *symptomatic*. It considers what perceived attributes of an act of construction would cause that act (or its agent, or its outcome) to be judged as creative$_S$; see Section 8.3.3 below. This attempts to capture

the idea that (strict) creativity is in the eye of the beholder, and to specify what observable factors would lead an observer to attribute creativity to the system. (See (Ritchie, 2007) and Colton (2008).)

The first two of these three perspectives (descriptive and causal) might be useful or inspiring to someone designing a potentially creative system, but they do not directly affect the question of evaluation. The third of these perspectives, symptomatic, lends itself more naturally to evaluation, since that activity is carried out by observing a computer system, and formulating a judgement about its behaviour. However, a causal account can also be relevant, in that the *perception* of the presence of a putative causal component can sometimes be treated as a symptom of creativity$_S$ (cf. Section 8.3.4 below). We shall, in this section, focus primarily on frameworks which contribute, perhaps indirectly, to a symptomatic perspective. This means that we shall have little to say about frameworks which are largely descriptive, including Wiggins' CSF.

Even if there were to be a well-developed symptomatic account (of creativity$_S$), an abstract characterisation of potential symptoms is not yet a concrete practical evaluation procedure. We will return to that aspect later (Sections 8.5 and 8.6).

8.3.2 Boden's Analysis

Although there have been many analyses of creativity within philosophy and psychology, the young discipline of CC has tended to focus on accounts from more recent decades, based in cognitive science or AI. By far the most influential of these is the work of Boden (1992, 1998, 2004), which has fed directly into much of the debate on evaluation in CC.

The most prominent idea that Boden develops is a three-way classification of creative activities into *combinational*, *exploratory*, and *transformational* (largely from descriptive and causal perspectives). Although this has attracted considerable attention within CC, we shall not examine it here, as this taxonomy is not fundamental to the question of evaluation in CC.

Instead, evaluation has been influenced by other concepts discussed by Boden, perhaps more symptomatic in their perspective. "Creativity is the ability to come up with ideas of artefacts that are *new, surprising and valuable*." Boden (2004, p. 1); see also Boden (1998, p. 347). In addition to these three concepts, Boden assumes another, which corresponds to what we have referred to as *acceptability*: "It is even more difficult to express. . . just what it is that we like about a Bach fugue, or an impressionist painting, than it is to recognize something as an acceptable member of one of those categories" Boden (1998, p. 354). For Boden, these four notions are relatively obvious aspects of the phenomenon of creativity, and her writings concentrate on explicating how such effects could be created (a "causal" account).

Novelty. Of these concepts, novelty receives the most analysis within Boden's work. It is emphasised as essential to creativity (which throughout Boden's writings

is what we have called "strict" creativity). A creative act is termed *P-creative* ("personal") if it manifests novelty within the experience of the creator, in the sense that the creator could have no knowledge of a comparable artefact prior to the creative act. In contrast, a creative act which produces an artefact which is novel with respect to the wider culture, even taking into account past history, can be *H-creative* ("historical"). Although Boden does not make this step, there is a natural generalisation to *P-novelty* (something totally new to the creator) and *H-novelty* (something new in a more absolute, historical and cultural sense). Boden observes that a cognitive model of creativity aims to describe P-creativity. Similarly, when considering novelty as part of the evaluation of a creative$_L$ system, we shall usually be concerned with P-novelty.

Quality. Boden discusses value/quality much less than novelty, but observes that "value" is an elusive notion, with human standards varying across cultures and historical periods (Boden, 2004, p. 10). Two of the earliest articles explicitly discussing evaluation within CC (Pearce & Wiggins, 2001; Ritchie, 2001) make it clear that quality is central, and this notion is now generally regarded within CC as both crucial and in need of rigorous examination.

Acceptability. Although Boden does not list "acceptability" as a primary symptom of creativity, it is central to her notion of "transformational" creativity, in which the boundaries of acceptability are altered. This distinction between "acceptable" and "valued" has been adopted within CC, appearing as a key element in the earliest formalisations (Ritchie, 2001; G. Wiggins, 2001).

Surprise. Boden (2004, pp. 2-3) distinguishes three types of surprise: where something is perceived as unexpected, based on statistics and past experience; where there is a realisation that an idea fits within a "style of thinking" in an hitherto unnoticed way; and where an idea previously thought to be impossible is recognised as possible. She states that these correspond, respectively, to combinational, exploratory, and transformational creativity. It is not entirely clear how Boden's notion of surprise, particularly the "combinational" variety, differs from novelty. If surprise is to be treated as separate from the other three concepts, then it must be different from low levels of acceptability, and from dissimilarity to previously encountered items. It may need some temporal component, in order to have "before" and "after" states. Maher et al. (2013) give a good discussion of the issues, and of some previous accounts of surprise, and describe an approach in which a linear regression model is used to make predictions. Macedo and Cardoso (2001) describe a computational mechanism of surprise which is similar to Boden's first subtype (combinational). Grace and Maher (2014) argue that *unexpectness* underlies not only surprise, but also novelty and transformation (See (Grace & Maher, 2017) for a detailed discussion of the relation of unexpectedness and surprise to creativity). These investigations of surprise focus on what we have classed as "internal" evaluation (Section 8.2.4).

The four properties listed in this subsection have some relevance to all the strands of CC described in Section 8.2.3 above. The fact that novelty, quality and acceptability have been put forward as possible facets of (strict) creativity, and have been widely

discussed from that perspective, makes them directly relevant to research with strong CC objectives. For a project with weak engineering CC objectives, the attainment of acceptability is an absolute minimum standard, with quality being extremely desirable, as is novelty. For weak scientific CC, acceptability is an indicator of the model's accuracy, whereas quality could act as a measure of additional precision, in the sense that high quality output may indicate that the model is focussed on the central core of the space of possibilities. Novelty is also relevant to weak scientific objectives, since it indicates a degree of *generality*; that is, the model can create exemplars other than those upon which its design was based (the "inspiring set" – see Section 8.3.3), and so has not simply been "fitted" to a particular dataset in an ad hoc way; cf. (Colton, Pease, & Ritchie, 2001; Gervás, 2017).

8.3.3 Some Symptomatic Criteria

Ritchie (2001) set out a set of formal criteria (revised and expanded in (Ritchie, 2007)) by which the behaviour of a creative$_L$ system could be assessed. (Convenient summaries, avoiding mathematical formulae, can be found in Jordanous (2012a, p. 250) and Gervás (2017)). This scheme assumed the availability of a small set of basic constructs: I (the *inspiring set*) which were the pre-existing artefacts upon which the design of the system was based; typ, a rating of the *typicality* of an output item (cf. "acceptability", Section 8.3.2); and val, a rating of the *value* of an output item (see Section 8.3.2). These were combined in various ways (e.g. averaging values, setting thresholds, intersecting sets) to formulate criteria applicable to a system's output. Novelty was not a primitive concept, but was taken into account in two ways: having a low value for typ, or being different from I.

Strictly speaking, the criteria were intended neither as necessary nor as sufficient conditions for a system to be creative$_S$, but more as ways of formulating precise statements about those aspects of the behaviour of a system which might be pertinent to CC (even with weak objectives). Some of the criteria were definitely not concerned with strict creativity, but more with the exploration of existing areas, of the sort we shall discuss in Section 8.5. In this way, applying all the criteria to a system (where applicable – the set I is often unknown) produces a profile of the system's behaviour, rather than a verdict about whether or not it has been creative$_S$. (As Bown (2014) observes, the title of Ritchie's 2007 paper is misleading over this point.) Despite this, there have been some cases where these criteria have been used to assess the creativity of systems; see (Ritchie, 2007) for a review, and (Gervás, 2017) for some examples. Ventura (2008) gives a detailed examination of some aspects of Ritchie's proposals, and Ritchie (2012) draws some parallels with the basic constructs of Wiggins' CSF.

These criteria are firmly intended as symptomatic, being devised wholly in order to set out a range of possible measures that an observer could apply in order to make precise, verifiable statements about a creative$_L$ system.

8.3.4 The Creative Tripod

Colton (2008) presents a *tripod* of three creativity-related "behaviours we require in our system": *skill*, *appreciation* and *imagination*. These are *necessary* conditions for a system to be creative$_S$. Of a painter (human or computational), Colton remarks: "Without skill, they would never produce anything; without appreciation, they would never produce anything of value; without imagination, at best they would only produce pastiches of other people's work." This emphasises that the justification for including these three concepts derives from what we called (Section 8.3.1) the "causal" perspective; that is, these elements contribute to an agent succeeding in being creative$_S$. However, Colton goes on to argue for using the tripod in what we have have called the "symptomatic" perspective: "... if we perceive that the software has been skillful, appreciative and imaginative, then ... the software should be considered creative. Without all three behaviours, it should not be considered creative." The latter sentence raises the subtle possibility that, rather than striving to specify when a system has been creative$_S$, it could be more realistic to specify sufficient conditions (on behaviour) for a system *not* to be considered creative$_S$. Colton, Pease, Corneli, Cook, and Llano (2014) argue for this as a preferred approach, speculating that ordinary usage of the terms "creative" and "uncreative" may allow a middle ground where something is judged to be neither of these; see also Colton et al. (2015, Section 4.1).

The increasingly inaccurately named tripod was later expanded to include further "words" which "people can meaningfully project" on to behaviours: *learning*, *intentionality*, *accountability*, *innovation*, *subjectivity* and *reflection* (Colton, Pease, Corneli, Cook, & Llano, 2014, p. 141). These were also to be used primarily from (in our terms) a symptomatic perspective. (More recently, the extended list of nine tripod components has been characterised as *essential behaviours* (Colton et al., 2015)). Pease and Colton (2011b) announced an intention to study a wide range of factors, including "... affect, analogy, appreciation, audience, autonomy, blending, community, context, curiosity, exploration, framing, humanity, humour, idea formation, imagination, intentionality, interaction, interpretation, knowledge, metaphor, novelty, obfuscation, personality, physicality, playfulness, problem solving, process, programming, search, surprise, transformation and trust" .

The tripod factors were specifically selected to enable judgements about when a system fails to be creative$_S$, and hence are directed at the strong CC agenda (Section 8.2.3). However, some of them might be of some use in the evaluation of CC with weak aims: for example, perceptions of skill or of imagination might be related to, or used instead of, judgements of greater acceptability or quality. Part of the difficulty in devising rigorous evaluations is that the use of ordinary words can lead to ambiguity, or to the specious appearance of differences. Bown (2014, p. 113), while accepting the intuitions which motivated the original Colton tripod, raises doubts about whether its terms have been, or could be, fully operationalised.

8.3.5 The IDEA Framework

A relatively detailed descriptive model is offered by the *IDEA* framework, which plots the development process of a creative system (Colton et al., 2011; Pease & Colton, 2011a). Amongst its many components, it includes a symptomatic perspective, as it allows for systematic ratings, by an "ideal audience" of each "act of creativity" in terms of two measures: *well-being*, indicating the extent to which the act is liked, and *cognitive effort*, representing the time the audience was prepared to spend attempting to understand the act. These two basic numerical ratings can then be combined mathematically to form secondary measures: Colton et al. (2011) offer formulae for *disgust*, *divisiveness*, *indifference*, *popularity*, and *provocation*. There are also definitions by which these secondary constructs are combined into further derived measures for *acquired taste*, *instant appeal*, *appeal splitting*, *opinion forming*, *shock*, *subversion*, and *triviality*. The IDEA model, and its companion FACE model, are "...based on theories of how creative work is received" , but are "...*inspired by*, rather than *models of*, human creativity." Pease and Colton (2011a, p. 75, italics in original)

8.3.6 Similarity

A generated item which was completely identical to some prior known artefact would generally be regarded as lacking novelty. Also, the generation of an item which was very similar to a previous artefact would not be seen as very novel. More generally, the key notion of "novelty" has a natural relationship to the concept of dissimilarity to previous exemplars. The idea of "similarity" has been proposed as an important aspect of evaluating the output of creative$_L$ systems (Gervás, 2002; Pease et al., 2001; Pereira, 2005; Pereira, Mendes, Gervás, & Cardoso, 2005; Ritchie, 2007). Although similarity to previous artefacts may, in itself, not indicate highly creative$_S$ behaviour, it could be a useful way of testing some of the other measures that have been proposed. For example, Haenen and Rauchas (2006) use this to approximate Ritchie's notion of "typicality" (Section 8.3.3), and some of Colton et al.'s stages of performance (Section 8.4.2 below) are stated in terms of similarity. Within a creative domain, it is not trivial to devise an automatic (i.e. objective) similarity metric which accurately captures the salient aspects of artefacts. Tearse, Mawhorter, Mateas, and Wardrip-Fruin (2011) attempt this for simple stories, and similarity is the key component of the creativity metric of Elgammal and Saleh (2015).

8.3.7 Vocabulary Analysis

Jordanous (2012a) analysed words and phrases used in a corpus of relevant literature, finding those which appeared significantly more often in creativity contexts, cluster-

ing these terms, and then attaching intuitively appropriate headings to the clusters. As a result, she recommends 14 concepts (including *active involvement and persistence*, and *social interaction and communication*) as a "base definition" in formulating an explicit statement of what would count as evidence of a system's creativity. (See also (Jordanous & Keller, 2016)).

Some words found during Jordanous's study were combined with other terms in a study by van der Velde, Wolf, Schmettow, and Nazareth (2015), which used various established techniques (word-association, card-sorting, cluster analysis, principal components analysis) to find, from human judgements, which words were most closely related to "creativity". Simplifying the results somewhat, this resulted in five dominant groups of words, labelled as: *skill*, *intelligence*, *novelty/innovation*, *emotion*, and *original*.

8.4 Stages of Development for a CC System

8.4.1 Formative versus Summative Evaluation

Evaluation, in the sense we are using the term, should not be confused with testing or debugging. Particularly in the early stages of developing a program, priority is often given – quite rightly – to fixing bugs, so that the program at least works. Once that basic level has been achieved, further improvements will take the program through successive versions, with occasional testing to monitor progress. It is then that genuine evaluation can be considered.

In an educational context, there is a long-standing distinction between *formative* and *summative* testing (of the skills or knowledge of learners). Formative testing is carried out during an educational course to give feedback to both teachers and students about progress. Summative testing is used as a final measure to determine what has been learned, and will typically occur at the end of a course. Traditional school homework exercises are generally formative, whereas qualifications such as diplomas and degrees are awarded on the basis of summative testing. We can borrow these two concepts, so that we have *formative evaluation* and *summative evaluation* (Jordanous, 2012a, p. 247).

Formative evaluation is more systematic than mere debugging, and is concerned with higher (and more interesting) aspects of a program than mere avoidance of errors. It examines whether a creative$_L$ system's output artefacts are becoming more acceptable, or of higher quality. It may also explore other aspects of the program's behaviour in which the programmer is interested, perhaps by varying some parameters (cf. Section 8.5). Although it should be done carefully and methodically, its main purpose is to give feedback to the program's designer/implementer, and so it need not be as rigorous as a scientific experiment.

Summative evaluation is carried out once the program has reached, usually by progressing through a series of versions, a level of performance where it is worth

reporting to the wider academic community. The aim of this type of evaluation is a thorough investigation of the capabilities of the current version of the program, in a way which obtains results that make meaningful and supported claims. This is CC's equivalent of the scientific experiment.

Although formative evaluation may discover some curious or interesting behaviour by the program, such findings are at most the material for anecdotes – they are not scientific findings to be reported. On the other hand, summative evaluation, if executed properly, is very much the source of reportable results. In addition, a summative evaluation may simultaneously offer ideas for future improvements (cf. Pearce and Wiggins (2007)); that is, it can also play a formative role.

A system design is possible in which evaluation by humans is part of a development cycle for improving the system's internal rating scheme (de Melo & Gratch, 2010; Gervás & León, 2010). This is not really summative evaluation, since it is not aimed at giving feedback on the system's performance, but is more an example of using humans to pseudo-automate a stage (internal evaluation, Section 8.2.4) in the creation process (cf. (Ritchie, 2008)).

8.4.2 Levels of Performance

In the course of developing a creative$_L$ system, there are a number of behaviours which could be used as milestones. (We shall ignore relatively low-level problems of implementation, and view the system as the embodiment of an abstract model of the particular domain, such as visual art, poetry or melody, which has certain parameters – Section 8.5 – and a well-defined output data type, *basic items* in the terminology of Ritchie (2007)). The significance of each of these attainments depends to some extent on the exact objectives of the research (see Section 8.2.3). Colton et al. (2011) propose some interesting behaviours which may occur during development.

One possible early stage is to find if the model can accurately describe known exemplars within the domain. That is, is there a way to run the program so that the output item is a known artefact? Although this may seem feeble in terms of strict creativity – since it clearly would not be novel – it at least shows that the model can create well-formed artefacts of a reasonable standard of quality; this is not trivial. Also, it is a significant step forward if the objectives are "weak" in the sense of Section 8.2.3. In the case where the known exemplars were in some way part of the system's development, and all the system's output is like this, the behaviour is labelled *developmental* by Colton et al. (2011, p. 93), and corresponds roughly to Criterion 9 in (Ritchie, 2007) (Section 8.3.3 above).

A slightly more advanced stage is to generate at least some items which deviate only slightly from known exemplars. If there is already a system which can generate exact replicas (and there is no guarantee that a system will ever pass through that stage), then it can be interesting to explore very small variations in the way the program is set up (cf. Section 8.5), to see what effect they will have; this may result in small variations in output. Or it may simply be that generating near-likenesses

of known exemplars is the first stable stage that the development reaches. Colton et al. distinguish two subcases of this: where the known exemplars were part of the system's development, they dub the situation *fine tuned*, and it is roughly Ritchie's Criterion 9b. Where the comparison artefacts were known more widely but were in no way part of the system's development, this is *reinvention* in Colton et al.'s scheme. This can be seen as adopting the distinction between "P-novelty" and "H-novelty" mentioned in Section 8.3.2 earlier (if we define novelty in terms of similarity rather than identity; cf. Section 8.3.6). That is, Colton et al.'s "fine-tuned" is where the artefact fails to be P-novel, and "reinvention" is where the artefact is P-novel but not H-novel.

Once again, the lack of novelty means that this may not count as creative$_S$ behaviour, but it is a useful intermediate stage, and it is of significant interest where the objectives are "weak".

This illustrates that the distinction between formative and summative evaluation (Section 8.4.1) may depend on the relationship between the behaviour evaluated and the objectives of the research. For a project with "weak" objectives, it would be reasonable to use the replication, or near-replication, of existing artefacts as a summative criterion, whereas with "strong" objectives, this would be more plausible as a formative measure.

Colton et al. suggest a further nuance on near-replication of known artefacts, using two different thresholds for degrees of similarity. Where the system produces an item which is not very similar to any known artefact, but is relatively similar to some known artefact which was not part of the system's development, this is *discovery*.

They also add two further levels of behaviour: *disruption* is where there is at least one output item which is not even relatively similar to any known item, and *disorientation* is where all output items meet this condition. Whether these are desirable candidates for evaluation criteria would depend very much on the surrounding theoretical agenda. For "weak" objectives, "disruption" might be a sign that the model did not characterise the known domain very well, and "disorientation" would be stronger evidence of a mismatch. However, with "strong" objectives, particularly if one is interested in *transformational creativity* (Boden, 1992, 1998, 2004; Ritchie, 2006), these behaviours would at least be interesting. Even with those aims, mere deviation from known exemplars, with no further qualifications (e.g. a high rating for quality) could not be counted as total success, although it might be a reasonable intermediate (formative) criterion in the quest for transformational behaviour.

Although Colton et al. present their categories as *stages* of the development of a system, they are really ways of categorising behaviour (like the criteria in Section 8.3.3). In particular, a system which generates several items in a run (cf. Section 8.5) could simultaneously display behaviour which was, by Colton et al.'s definitions, "fine tuned", "reinvention", "discovery" and "disruption", if there were output items which qualified under these definitions.

8.5 Organising Evaluation Runs

Although the testing of a creative system can be arranged in many different ways (see Section 8.6.3 below), most schemes involve the generation of a set of artefacts for use in the evaluation. There are various different ways that this can be approached, depending on the aims of the research and the functionality of the system. Generally, summative evaluation requires design decisions about two major aspects: how to initialise the program, and how to select output items for formal assessment. We shall now consider these two phases in more detail, but first we have to elaborate on the notion of a "parameter".

As emphasised by the formalisation in Wiggins' CSF (see Section 8.2.2), a creative$_L$ system usually explores a space of possibilities, sometimes referred to as a *conceptual space*. Such spaces can be viewed as having multiple dimensions, as set out by Gärdenfors (Gärdenfors, 1990, 2004; Gärdenfors & Williams, 2001). A dimension need not be ordered (as it would be in a traditional representation of three-dimensional space), but is simply a set of possible values. We will refer to these sets as *artefact space dimensions*. A choice of values for these dimensions then specifies a location in the space of possible artefacts, and hence the dimensional structure can be used to organise a search within the set of artefacts. A clear example of this is the way that Isaksen, Gopstein, Togelius, and Nealen (2015) use a genetic algorithm to explore a space containing possible games of a particular genre; they refer to these controlling values as "parameters". Sometimes, a program will allow the experimenter to specify values, or sets of values, for some of the artefact space dimensions, thereby confining the search to a particular area of the space.

Wiggins emphasises that the behaviour of a creatively searching program can be seen as following some form of search procedure which defines the trajectory of the search through the conceptual space. Some CC programs may allow the experimenter to specify settings which affect this computational behaviour. We shall refer to these as *program control settings*.

Thus there are two sorts of parameter which the experimenter may be able to adjust when initialising a program before an evaluation run: artefact space dimensions, and program control settings. Which of these are actually available will depend on the details of the particular program, since values are sometimes hard-wired into the software, or left for the program to vary freely. Those which are available to the experimenter we shall call *initial parameters*,

Any aspect of the program which is modifiable by the experimenter and which affects its output or behaviour counts as an initial parameter, in this sense. For example, the JAPE riddle generator (Binsted, 1996) had various types of rules (schemas, templates, etc.), and by default a program run was an exhaustive search of all possible instantiations of all possible rule combinations. However, if the list of available schemas were restricted, then a smaller space would be searched. Thus, the set of available schemas was in effect an artefact space dimension, and also an initial parameter, as were the other rule sets, or the choice of which lexicon to use. (This was made explicit in later implementations of JAPE's algorithms (Manurung et al., 2008)).

When generating artefacts for the purposes of evaluation, a suitably large set of output items will usually be needed. This does not necessarily require multiple runs of the program, since it is often possible to organise a program so as to return, in a single run, many elements from within the artefact space. However, it can also be interesting to execute several program runs, if there are non-trivial initial parameters which can be modified. (In the unlikely event that a system has no initial parameters, then the only course of action is simply to run the program as it is.)

This means that making an run (or runs) of the program for the purposes of evaluation can be seen as investigating how choices of initial parameters lead to particular subareas of artefact space. The two facets of experiment design mentioned above – initialisation and output selection – are the ways in which the experimenter implements this investigation. A significant part of the design of a systematic evaluation is therefore the definition of these two choice policies: setting initial parameters, and selecting output artefacts. There are a range of policies, most of which are applicable, perhaps in slightly modified form, either to initial parameter setting or to artefact selection; these include the following:

Exhaustive. For initial parameters, this is the mode described above for the original JAPE system: search all possible values of the parameter. Where the space of values makes this wholly intractable (for example, a real-valued parameter within some numeric range), then a near substitute would be to sample the parameter space uniformly at intervals across the space. For artefact selection, this mode means putting every output item into the formal evaluation; this is often infeasible because of large quantities of items.

Neighbourhood. For initial parameters, this involves finding values which yield interesting results, and then making minor modifications to them (cf. Section 8.4.2). This would be appropriate as part of formative evaluation, or when the objectives are "weak". It is less clear that there is a useful policy of this sort in artefact space, but it is conceivable that it might be useful when calibrating or validating an evaluation method.

Representative sampling. If the experimenters have some prior hypotheses about the relationship between some initial parameter values and the outputs in artefact space, or between different subareas within artefact space, then this should guide the choice of initial parameter values and also the selection of output items. If there is a conjecture that area A of the initial parameter space will yield artefacts with better values than area B will, then these two areas should be explored. Or if there is a hypothesis that a certain area of artefact space is particularly highly valued, that should guide artefact selection (including, for comparison, items from outside that area). For example, if there is a hypothesis that paintings with bright colours are deemed more creative than those with less saturated colours, then the dataset should include clear exemplars of both of these types of artefact.

Random. If the exhaustive method would be impractical, and there are no prior principles or evidence to suggest using any of the more structured sampling methods listed above, then the default is to set the initial parameters, or to select

output, randomly. ("Random" does not mean "arbitrary" – a genuinely random method should be used to select values.)

Curation. If previous runs, perhaps as part of formative evaluation, have shown that particular initial-parameter values result in "good" results, there can be a temptation to use these values in an evaluation which is summative (i.e. intended for publication). This is also known as *fiddling the results*. Similarly, the researcher may subjectively choose those output items which intuitively seem to be "good", a practice referred to by its supporters as *curation*, while its critics prefer the term *cherry-picking*. (Llano, Colton, Hepworth, and Gow (2016) suggest using subjective filtering of the output items, by the research team, as a *formative* evaluation method. They define the *curation coefficient* as the proportion of the program's output items which are subjectively deemed to be sufficiently well-formed to be worth assessing.)

The presentation above is not intended to imply that the same policy must be adopted for both initial-parameter setting and output selection; these are usually independent design decisions.

8.6 Ratings and Measurement

8.6.1 Quality or Creativity?

In other areas of AI, evaluation is essentially concerned with how good the overall performance of the computer system is. For CC systems, the exact meaning of "good performance" depends to some extent on the goals of the research. For a project with "weak" objectives, it is likely that "acceptability" and "quality", in the sense of Section 8.3.2, are the central considerations. The experimenters, in order to check the accuracy of their domain model, will wish to check if it results in the creation of acceptable and high-quality exemplars within the domain. Some evaluations in CC follow this path, even if they do not always explicitly declare weak objectives.

Where the objective is "strong" (i.e. in pursuit of strict creativity), the experimenter has to decide what working definition of creative$_S$ is to be adopted, and, following from this, which aspects of the program's performance should be measured to gather evidence. It would be possible to try to measure "creativity" directly, but this could at best result in a simple labelling of program behaviour as being seen as more or less creative, without casting light on why or how this perception of creativity arose. This has led to proposals for what we earlier called "symptomatic" criteria for creativity (Section 8.3.1). At least some of these (Sections 8.3.3 and 8.3.5) share a methodological strategy: start from a small set of *primary measures* which can be directly determined (e.g. using human judgements) and then combine these values into *secondary measures* which express more complex and subtle properties of the behaviour of a potentially creative system. In Ritchie's scheme, the primaries are membership of the "inspiring set", and ratings for "typicality" and "value", and in

the IDEA framework, the primary measures are "well-being" and "cognitive effort". In many cases, including these two, the assumption is that the primary measures can be assessed using human judges (Section 8.6.4), although in principle there could be a framework in which the primaries could be measured objectively (Section 8.6.6),

An interesting question is: what is a theoretically interesting set of primary measures which are workable in practice? Usually, this has led to some variant of "quality" featuring in evaluations of strict creativity. Given the centrality of quality in evaluations aimed at "weak" objectives, this means that quality evaluation is at the heart of all CC evaluation. Jordanous (2011) found that many CC projects base their evaluations primarily on quality.

Where the research is concerned to explore "transformational" creativity – which we are not examining here – the challenge to define persuasive evidence is even greater (Ritchie, 2006).

8.6.2 Effects

In some non-CC areas, evaluation uses *task-based* measures: a system is successful to the extent that its behaviour or output facilitates the performance of some task by a human subject. Although there may be areas of CC where this could be adopted, most creative$_L$ systems are not intended to support a particular practical task. For a creative$_L$ system, perhaps the nearest counterpart to measuring influence on task performance would be some measure of the user's emotional state, since many creative artefacts are, broadly speaking, intended to influence the emotions of the experiencer. This leads to two large questions which are still relatively unexplored within the CC community: what emotional effects should particular artefacts induce in the user, and how could we best measure these effects?

The measurement of the effect of computer systems on emotions and mood is of interest within the field of "affective computing" (Picard, 2000; van der Sluis & Mellish, 2008). A number of methods are available from other disciplines, including questionnaires (Watson, Clark, & Tellegen, 1988), physiological factors (Kassam & Mendes, 2013), body language (Aviezer, Trope, & Todorov, 2012) and facial expression (Bartlett, Hager, Ekman, & Sejnowski, 1999), but the issue is still not trivial. Also, it is not always obvious exactly which emotional effect(s) would be apposite for a particular creative$_L$ system. (There may be exceptions: for humour, amusement is presumably the predicted response.)

Ideally, this kind of evaluation would measure actual emotional responses, as opposed to asking judges to rate the artefacts for emotional content or impact (as done, for example, by Monteith, Martinez, and Ventura (2010), and de Melo and Gratch (2010)).

8.6.3 Naturalistic Setting

Typically, evaluations are viewed as something which happens under controlled conditions, so that the experience for the human participants can be carefully designed and restrictions can be put on the factors which are involved. It is also worth considering a more "natural" form of evaluation. In this, the participant encounters the material to be assessed in some context which is as appropriate as possible, and then reacts or interacts within that setting.

Outside CC, there have been arguments that this is a suitable way to assess the usability of complex computer systems: let the user experience the system in a real setting, and then observe what happens. The information gathered in this way is much more qualitative than in typical laboratory contexts. Some attempt can be made to structure the observation, using guidelines and questionnaires (cf. the use of experts; see Section 8.6.4 below). Shneiderman and Plaisant (2006) outline very detailed ways in which *ethnographic* methods could be used, although they are not concerned with creative programs.

For CC, the equivalent arrangement would be to place generated artefacts in a setting where artefacts of that genre are naturally encountered. It is possible, for example, to place computer art in a public gallery (Colton & Pérez-Ferrer, 2012; Colton & Ventura, 2014), to play computer music at concerts (Eigenfeldt, Burnett, & Pasquier, 2013), to serve soup made with computer-generated recipes (Colton & Ventura, 2014) or to allow users to play a generated game (Cook, Colton, & Gow, 2013). This can then generate reactions from the public (e.g. purchasing artworks, applauding music). Such feedback is more natural than a questionnaire completed in a laboratory, but it is harder to extract quantitative scientific results from it. Also, as Jordanous (2012a, p. 253) observes, there might be some ambiguity about whether the naturalistic reactions are to (strict) creativity or to quality.

There are two facets of this style of evaluation where there is a potential choice between control and naturalism: there is the *experience*, in which the participant encounters the creative system and/or its outputs, and there is the *measurement* phase. For example, Eigenfeldt et al. (2013) presented musical artefacts naturalistically, but collected feedback in a conventional academic manner. In contrast, an observational methodology (cf. the Shneiderman and Plaisant study cited above) would allow the assessments to be collected without the participants experiencing intervention from the experimenter.

Bown (2014) argues for the use of the *interaction design* methodology in evaluating creative systems, and Kantosalo, Toivanen, and Toivonen (2015) describe using this technique to evaluate a poetry-generator. Although this is not a wholly naturalistic approach, since it is based on structured use of the creative system and systematic collection of feedback, it could be seen as a step in the direction of increased realism within evaluation.

8.6.4 Use of Judges

Where a system generates artefacts in a domain which is subject to objective measures of success, such as the design of electronic circuit layouts, measuring "quality" may be possible without human involvement, but that is not the typical situation within CC. It is much more common for a creative$_L$ system, more or less by definition, to create artefacts which can be assessed only subjectively. In such circumstances, human judgements have to form a part of the evaluation. Where naturalistic evaluation is not realistic (which currently is the usual situation), this means some form of structured, controlled presentation, to human judges, of system behaviour or (more normally) generated artefacts. Although it might be acceptable for the experimenters to assess their own system's constructions as part of early formative evaluation, the need to avoid conflict of interest and to achieve a degree of rigour means that other persons should be involved in any summative judging.

There are, broadly, two distinct roles that can be given to the judges; we shall label them as *sample audiences* or *expert annotators*:

Sample audiences. A relatively large number of participants are asked to experience the system's behaviour in some way, very often by being shown example output items (e.g. computer-generated stories) and are asked to judge them, for example on a set of Likert scales. This audience is taken as being representative of the population of people at whom the creative work is targeted. This is directly comparable to a psychology experiment, and the usual guidelines apply (Section 8.6.7). Specifically, the judges should be chosen to be suitable (e.g. native speakers if linguistic verdicts are needed), there should be a large enough pool of participants to allow statistical testing, and the audience do not need to be told the true aim of the testing (and it is usually better if they are not).

Expert annotators. In this case, a small number of knowledgeable people study the data and record their verdicts about whatever features are of interest in the investigation. The amount of agreement between the experts can be gauged using various statistical tests (Krippendorf, 1980; Neuendorf, 2002). This sort of approach has been used for many years in fields such as machine learning, where the aim is to produce a dataset with an agreed labelling which can then be used as training or test data. Also, expert mark-up is used in the social sciences when a research hypothesis is to be tested across some dataset. The latter scenario is closer to CC evaluation, where methods of measuring agreement could be used to provide more solid verdicts on creative output, for example amongst a small panel of art critics. This approach sometimes uses part of the research team as the experts, but this is far from ideal – it is better if external annotators can be recruited.

Describing this as a dichotomy is an oversimplification, as some intermediate variations are possible in how judging is arranged. Also, the distinction made here is logically distinct from the matter of the judges' level of domain expertise, which can vary greatly across studies (Jordanous, 2012a) and within a study (Eigenfeldt &

Pasquier, 2011). For example, the judges used by Pearce and Wiggins (2007) had strong domain expertise, but were not briefed on the origins of the material they were rating, and their role within the experiment was roughly what we have called "sample audiences". (That study was based on the *consensual assessment technique* (Amabile, 1996).)

Whatever the judging arrangement, it is important to consider carefully what to ask of the judges, an issue for which there is well-established practice within psychology (cf. Section 8.6.7).

8.6.5 The Turing Test

It might seem, inspired by Turing (1950) and the subsequent development of the "Loebner Prize" (n.d.), that a useful measure of success for a creative$_L$ system would be for its artefacts to be indistinguishable from human creations in the same domain, when encountered in a suitable setting. This has come to be referred to as the *Turing Test* (TT). Pease and Colton (2011b) present a detailed criticism of the careless use of this idea.They point out that some discussions of the TT actually envisage something which omits the essence of the TT, either by not being interactive (a crucial aspect of the TT), or by not being tests of discrimination. Pease and Colton also agree with other critics that adopting TT evaluation could lead to excessive concentration on superficial tricks specifically aimed at passing the test. More significantly, they argue that a genuine TT is not an appropriate evaluation method for CC, as it primarily rewards near-replication of human behaviour (and hence of human-created artefacts).

Pease and Colton explicitly note that they are, throughout their critique, considering the evaluation of (strict) creativity, not merely value/quality. That is, the methodological setting is that of "strong" objectives, rather than "weak" objectives (Section 8.2.3). With weak objectives, quality would be of central interest, and similarity to human work would be relatively desirable (cf. Section 8.3.6). Hence, their arguments do not mean that comparison between computer-generated artefacts and human-generated artefacts – which does not in itself constitute a TT – has no place in evaluation within CC. Such a comparison can be extremely useful in setting up a frame of reference for various measures, particularly where quality (which Pease and Colton describe as "an essential criterion for creativity") is to be assessed. Any rating scales used in evaluation have to be calibrated, in order that there is some objective notion of which ratings are bad/average/good for the domain in question. This can be done by including human-created artefacts, and possibly other controls, such as randomly generated items, alongside the computer-generated items (cf. (Binsted, Pain, & Ritchie, 1997; Pearce & Wiggins, 2007; Peinado & Gervás, 2006)). Also, concealing the fact that some of the items are computer-created may be useful to avoid any bias that the judges may have regarding computer generated works (Moffat & Kelly, 2006; Mumford & Ventura, 2015; Norton et al., 2015).

8.6.6 Could It Be Automated?

In some fields of computer science and AI, automated evaluation, without the intervention of human judges, is already established. Typically, a program's performance is assessed by running it on some benchmark data and comparing the results with some "gold standard" representing optimal performance. The use of a single correct solution is not usually applicable in CC (although, as with task-based evaluation, this might be acceptable for some of the more structured or formalised areas). Here we are considering only the central question of possible automated rating of a creative system's behaviour, particularly its generated artefacts, and are leaving aside the separate question of whether automatic testing could be devised for some secondary CC-related aspect of a program (for example, Agustini and Manurung (2012) describe a program which tries to automatically extract templates from a corpus of computer-generated riddles).

In some non-CC areas, automatic evaluation methodologies have been set up even where there is not a single definitive benchmark of correctness. For example, this process has become particularly influential in recent decades within statistical machine translation. In that subfield, the gold standard is a set of translations, by human translators, of some benchmark texts. Different translation programs are then compared by seeing how similar their translations are to the standard human translations (Papineni, Roukos, Ward, & Zhu, 2002). Thus "similarity" is at the heart of this methodology (cf. Section 8.3.6).

Where a creative system includes, as part of its computations, an explicit test of the value/quality of items it is generating, then this might be labelled as "automatic evaluation". However, this is what we earlier categorised as *internal* evaluation, not the *external* evaluation which is our topic (Section 8.2.4). That is, in this chapter we are considering methods which might be used, *outside the model*, to estimate the overall success of a (potentially) creative system. In order to automate this external type of evaluation, it would be necessary to devise an algorithmic evaluation of the system's output (or behaviour) which was so accurate that the summative evaluation of the system could be delegated to the algorithm without human oversight. (Such an algorithm would have to be validated first, to ensure it was accurate; that is, it would itself have to be evaluated, presumably not automatically.) If this arrangement were achieved, then the algorithm would embody a hugely successful model of the domain in question (or maybe even of creativity in general). That would then prompt the query: why is this wonderful algorithm not implemented as part of the program, instead of being kept outside the system as part of the judging process? Any evaluation mechanism which was both genuinely automatic and highly effective would find a natural location *inside* the generation model.

Elgammal and Saleh (2015) have devised a sophisticated algorithm for rating creativity, but have not validated its results against independent human judgements (in view of "the absence of ground truth for creativity").

Pérez y Pérez et al. (2011) and Pérez y Pérez and Ortiz (2013) claim to have devised automated methods, apparently for external evaluation, of novelty and of interestingness in their story generator. The algorithms operate on internal knowledge

structures within their program (not the finished artefacts), the novelty tester has no validation against human judgements and the interestingness tester is only sketchily compared with human verdicts. All this means that the case has not been made that these algorithms are authoritative enough to take over as summative judges on the authors' story generator. And, as remarked above, if the algorithms are even moderately effective, it is unclear why the authors would not use them to enhance the generator.

8.6.7 Experimental Design

As remarked in Section 8.6.4 above, where evaluation involves the collection of human judgements, it has much in common with experimental psychology. There are many methods within psychology for gathering human responses; as noted earlier, these include questionnaires, reaction times and eye-tracking. With each data collection method, there is a wealth of sophisticated methodology concerning its administration. In addition, methods for statistical analysis of collected data are extremely well developed and documented. Research in CC therefore has no excuse for performing evaluations which collect or analyse human responses in ways which are not systematic and rigorous. Good practice can be borrowed directly from textbooks in experimental psychology.

Jordanous (2011, 2012a, 2017) proposes a set of standard evaluation guidelines for CC (with strong objectives), which in essence say: decide what you mean by creativity in the context of your system, specify how you will measure this, and implement these measures. In its barest form, this (excellent) advice leaves open the decision about the "key components of creativity", although Jordanous also offers a catalogue of possibilities (see Section 8.3.7).

8.7 Conclusions

The question of how best to evaluate a potentially creative system is not easy. Since 2001 there has been a continuing debate on the issue, often quite abstract and philosophical, and this is a welcome indication that there is real concern within the community. However, evaluation has not yet become a routine part of a CC project. A relatively recent survey of the evaluation of CC systems concluded that

> . . . evaluation of computational creativity is not being performed in a systematic, rigorous manner, but instead current practice is variable and somewhat ad hoc across the field. . . no evaluation methodology has been accepted as standard for evaluating and comparing the creativity of computational creativity systems . . . there is no consensus within the computational creativity community about which methodology to adopt. . . .
>
> Jordanous (2012a, p. 252)

Jordanous (2014, 2017) moves the debate into meta-evaluation: examining the strengths and weaknesses of different evaluation methods. It may be difficult to take such comparisons very far, given the underdeveloped nature of existing evaluation techniques.

We have argued the following points:

- The word "creative" is used in two senses: the *loose* sense is used to denote computer systems which operate in particular areas, typically of an artistic nature, and the *strict* sense is used to describe systems or accomplishments which meet a particular standard of "genuine creativity".
- Work in CC can have various goals. In particular, some projects pursue *weak* objectives, in which the primary or sole interest is the modelling of a specific domain, whereas others aim for *strong* objectives, in which success is equated with the achievement of strict creativity.
- There is no consensus about which aspects of a computer system's behaviour should be considered when assessing its possible strict creativity. However, the concepts of *novelty* and *quality* recur, sometimes in slightly different terms.
- In deciding which evaluation methods to use, it is important to take into account the stage of development of the computer system, as *summative* evaluation has more stringent requirements than *formative* evaluation.
- There is a range of ways of organising how test runs are set up during evaluation, with the setting of input parameters and the selection of output items being crucial.
- There are some established ways of having computer-generated artefacts rated by humans, but this is still an area where new methods are being explored. Good practice from experimental psychology should be followed wherever it is relevant.

Although the sheer variety of work in CC means that no single evaluation method can apply to all research, it would be desirable if the field could reach a stage where there is a known set of recognised and well-defined methodologies, so that every project report could include a succinct section describing an evaluation which used standard methods.

Acknowledgements I would like to thank Christian Guckelsberger for helpful comments on an earlier draft, and the anonymous reviewers for useful feedback. This chapter was written with the support of the Universities Superannuation Scheme Ltd, UK.

References

Agustini, T., & Manurung, R. (2012). Automatic evaluation of punning riddle template extraction. In M. L. Maher, K. Hammond, A. Pease, R. P. y Pérez, D. Ventura, & G. Wiggins (Eds.), *Proceedings of the 3rd international conference on computational creativity, ICCC-2012* (pp. 134–139).

Amabile, T. M. (1996). *Creativity in context*. Boulder, Colorado: Westview Press.

Aviezer, H., Trope, Y., & Todorov, A. (2012). Body cues, not facial expressions, discriminate between intense positive and negative emotions. *Science*, *338*, 1225–1229.

Bartlett, M. S., Hager, J. C., Ekman, P., & Sejnowski, T. J. (1999). Measuring facial expressions by computer image analysis. *Psychophysiology*, *36*, 253–263.

Bénabou, R., & Tirole, J. (2003). Intrinsic and extrinsic motivation. *Review of Economic Studies*, *70*, 489–520.

Binsted, K., Pain, H., & Ritchie, G. (1997). Children's evaluation of computer-generated punning riddles. *Pragmatics and Cognition*, *5*(2), 305–354.

Binsted, K. (1996). *Machine humour: An implemented model of puns* (Doctoral dissertation, University of Edinburgh, Edinburgh, Scotland).

Boden, M. A. (1992). *The creative mind*. First published 1990. London: Abacus.

Boden, M. A. (1998). Creativity and Artificial Intelligence. *Artificial Intelligence*, *103*, 347–356.

Boden, M. A. (2004). *The creative mind* (2nd). First edition 1990. London: Routledge.

Bown, O. (2014). Empirically grounding the evaluation of creative systems: Incorporating interaction design. In S. Colton, D. Ventura, N. Lavrač, & M. Cook (Eds.), *Proceedings of the 5th international conference on computational creativity, ICCC 2014*.

Cohen, P. R. (1995). *Empirical methods for artificial intelligence*. Cambridge, Mass: MIT Press.

Colton, S. (2008). Creativity versus the perception of creativity in computational systems. In *Papers from the AAAI spring symposium on creative systems* (pp. 14–20). Technical Report SS-08-03. Stanford, California.

Colton, S. (2012). The Painting Fool: Stories from building an automated painter. In J. McCormack & M. d'Inverno (Eds.), *Computers and creativity* (Chap. 1, pp. 3–38). Berlin: Springer-Verlag.

Colton, S., Charnley, J., & Pease, A. (2011). Computational creativity theory: The FACE and IDEA descriptive models. In D. Ventura, P. Gervás, D. Fox Harrell, M. L. Maher, A. Pease, & G. Wiggins (Eds.), *Proceedings of the 2nd international conference on computational creativity, ICCC 2011* (pp. 90–95).

Colton, S., Cook, M., Hepworth, R., & Pease, A. (2014). On acid drops and teardrops: Observer issues in computational creativity. In *Proceedings of the AISB symposium on AI and philosophy*.

Colton, S., Pease, A., Corneli, J., Cook, M., Hepworth, R., & Ventura, D. (2015). Stakeholder groups in computational creativity research and practice. In T. R. Besold, M. Schorlemmer, & A. Smaill (Eds.), *Computational creativity research: Towards creative machines* (pp. 3–36). Paris: Atlantis Press.

Colton, S., Pease, A., Corneli, J., Cook, M., & Llano, T. (2014). Assessing progress in building autonomously creative systems. In S. Colton, D. Ventura, N. Lavrač, & M. Cook (Eds.), *Proceedings of the 5th international conference on computational creativity, ICCC 2014*.

Colton, S., Pease, A., & Ritchie, G. (2001). The effect of input knowledge on creativity. In R. Weber & C. G. von Wangenheim (Eds.), *Case-based reasoning: Papers from the workshop programme at ICCBR 01*, Vancouver.

Colton, S., & Pérez-Ferrer, B. (2012). No photos harmed/growing paths from seed - an exhibition. In P. Asente & C. Grimm (Eds.), *Joint symposia on sketch-based interfaces and modeling, non-photorealistic animation and computational aesthetics (NPAR)* (pp. 1–10). Annecy, France: ACM.

Colton, S., & Ventura, D. (2014). You can't know my mind: A festival of computational creativity. In S. Colton, D. Ventura, N. Lavrač, & M. Cook (Eds.), *Proceedings of the 5th international conference on computational creativity, ICCC 2014* (pp. 351–354).

Colton, S., & Wiggins, G. (2012). Computational creativity: The final frontier? In L. de Raedt, C. Bessiere, D. Dubois, & P. Doherty (Eds.), *Proceedings of the 20th European conference on artificial intelligence* (pp. 21–26). Amsterdam: IOS Press.

Cook, M. (2015). Make something that makes something: A report on the first procedural generation jam. In H. Toivonen, S. Colton, M. Cook, & D. Ventura (Eds.), *Proceedings of the 6th international conference on computational creativity, ICCC 2015* (pp. 197–203).

Cook, M., Colton, S., & Gow, J. (2013). Nobody's a critic: On the evaluation of creative code generators – a case study in videogame design. In M. L. Maher, T. Veale, R. Saunders, & O. Bown (Eds.), *Proceedings of the 4th international conference on computational creativity, ICCC 2013* (pp. 123–130).

Corneli, J., Jordanous, A., Sheppard, R., Llano, M., Misztal, J., Colton, S., & Guckelsberger, C. (2015). Computational poetry workshop: Making sense of work in progress. In H. Toivonen, S. Colton, M. Cook, & D. Ventura (Eds.), *Proceedings of the 6th international conference on computational creativity, ICCC 2015* (pp. 268–275).

de Melo, C. M., & Gratch, J. (2010). Evolving expression of emotions through color in virtual humans using genetic algorithms. In D. Ventura, A. Pease, R. Pérez y Pérez, G. Ritchie, & T. Veale (Eds.), *Proceedings of the 1st international conference on computational creativity, ICCC 2010* (pp. 248–257). Department of Informatics Engineering, University of Coimbra.

Eigenfeldt, A., Bown, O., Brown, A. R., & Gifford, T. (2016). Flexible generation of musical form: Beyond mere generation. In F. Pachet, A. Cardoso, V. Corruble, & F. Ghedini (Eds.), *Proceedings of the 7th international conference on computational creativity, ICCC 2016* (pp. 264–271). Paris.

Eigenfeldt, A., Burnett, A., & Pasquier, P. (2013). Evaluating musical metacreation in a live performance context. In M. L. Maher, T. Veale, R. Saunders, & O. Bown (Eds.), *Proceedings of the 4th international conference on computational creativity, ICCC 2013* (pp. 140–144).

Eigenfeldt, A., & Pasquier, P. (2011). Negotiated content: Generative soundscape composition by autonomous musical agents in coming together: Freesound. In D. Ventura, P. Gervás, D. Fox Harrell, M. L. Maher, A. Pease, & G. Wiggins

(Eds.), *Proceedings of the 2nd international conference on computational creativity, ICCC 2011* (pp. 27–32).

Elgammal, A., & Saleh, B. (2015). Quantifying creativity in art networks. In H. Toivonen, S. Colton, M. Cook, & D. Ventura (Eds.), *Proceedings of the 6th international conference on computational creativity, ICCC 2015* (pp. 39–46).

Gallie, W. B. (1956). Essentially contested concepts. *Proceedings of the Aristotelian Society*, *56*, 167–198.

Gärdenfors, P. (1990). Induction, conceptual spaces and AI. *Philosophy of Science*, *57*, 78–95.

Gärdenfors, P. (2004). Conceptual spaces as a framework for knowledge representation. *Mind and Matter*, *2*(2), 9–27.

Gärdenfors, P., & Williams, M.-A. (2001). Reasoning about categories in conceptual spaces. In *Proceedings of the 14th international joint conference on artificial intelligence, IJCAI 2002* (pp. 385–392). Morgan Kaufmann.

Gervás, P. (2002). Exploring quantitative evaluations of the creativity of automatic poets. In C. Bento, A. Cardoso, & G. Wiggins (Eds.), *2nd workshop on creative systems, approaches to creativity in artificial intelligence and cognitive science, ECAI 2002*, Lyon, France.

Gervás, P. (2009). Computational approaches to storytelling and creativity. *AI Magazine*, *30*(3), 49–62.

Gervás, P. (2017). Exploring quantitative evaluations of the creativity of automatic poets. In T. Veale & F. A. Cardoso (Eds.), *Computational creativity: The philosophy and engineering of autonomously creative systems*. Computational Synthesis and Creative Systems. Springer International Publishing.

Gervás, P., & León, C. (2010). Story generation driven by system-modified evaluation validated by human judges. In D. Ventura, A. Pease, R. Pérez y Pérez, G. Ritchie, & T. Veale (Eds.), *1st international conference on computational creativity, ICCC 2010* (pp. 85–89). Lisbon.

Grace, K., & Maher, M. L. (2014). What to expect when you're expecting: The role of unexpectedness in computationally evaluating creativity. In S. Colton, D. Ventura, N. Lavrač, & M. Cook (Eds.), *Proceedings of the 5th international conference on computational creativity, ICCC 2014* (pp. 120–128).

Grace, K., & Maher, M. L. (2015). Specific curiosity as a cause and consequence of transformational creativity. In H. Toivonen, S. Colton, M. Cook, & D. Ventura (Eds.), *Proceedings of the 6th international conference on computational creativity, ICCC 2015* (pp. 260–267).

Grace, K., & Maher, M. L. (2017). Expectation-based models of novelty for evaluating computational creativity. In T. Veale & F. A. Cardoso (Eds.), *Computational creativity: The philosophy and engineering of autonomously creative systems*. Computational Synthesis and Creative Systems. Springer International Publishing.

Haenen, J., & Rauchas, S. (2006). Investigating artificial creativity by generating melodies, using connectionist knowledge representation. In *Proceedings of 3rd joint workshop on computational creativity, ECAI* (pp. 33–38). Riva del Garda, Italy.

Harmon, S. (2015). Figure8: A novel system for generating and evaluating figurative language. In H. Toivonen, S. Colton, M. Cook, & D. Ventura (Eds.), *Proceedings of the 6th international conference on computational creativity, ICCC 2015* (pp. 71–77).

Haugeland, J. (Ed.). (1981). *Mind design: Philosophy, psychology, artificial intelligence*. Cambridge, MA.: MIT Press.

Hayes, P. J. (1975). Nine deadly sins. *AISB European Newsletter*, 15–17.

Hayes, P. J., & Ford, K. (1999). Old sins & new confessions. *AI Magazine*, *20*(2), 128.

Isaksen, A., Gopstein, D., Togelius, J., & Nealen, A. (2015). Discovering unique game variants. In *Proceedings of 1st workshop on computational creativity & games*.

Jennings, K. E. (2008). Developing creativity: Artificial barriers in artificial intelligence. In P. Gervás, R. Pérez y Pérez, & T. Veale (Eds.), *Proceedings of the 5th international joint workshop on computational creativity* (IT/2008/2, pp. 1–10). Technical Report. Universidad Complutense de Madrid.

Jordanous, A. (2011). Evaluating evaluation: Assessing progress in computational creativity research. In D. Ventura, P. Gervás, D. Fox Harrell, M. L. Maher, A. Pease, & G. Wiggins (Eds.), *Proceedings of the 2nd international conference on computational creativity, ICCC 2011* (pp. 102–107).

Jordanous, A. (2012a). A standardised procedure for evaluating creative systems: Computational creativity evaluation based on what it is to be creative. *Cognitive Computation*, *4*(3), 246–279.

Jordanous, A. (2012b). *Evaluating computational creativity: A standardised procedure for evaluating creative systems and its application* (Doctoral dissertation, Department of Informatics, University of Sussex).

Jordanous, A. (2014). Stepping back to progress forwards: Setting standards for meta-evaluation of computational creativity. In S. Colton, D. Ventura, N. Lavrač, & M. Cook (Eds.), *Proceedings of the 5th international conference on computational creativity, ICCC 2014* (pp. 129–136).

Jordanous, A. (2017). Evaluating evaluation: Assessing progress and practices in computational creativity research. In T. Veale & F. A. Cardoso (Eds.), *Computational creativity: The philosophy and engineering of autonomously creative systems*. Computational Synthesis and Creative Systems. Springer International Publishing.

Jordanous, A., & Keller, B. (2016). Modelling creativity: Identifying key components through a corpus-based approach. *PLoS ONE*, *11*(10).

Kantosalo, A., Toivanen, J. M., & Toivonen, H. (2015). Interaction evaluation for human-computer co-creativity: A case study. In H. Toivonen, S. Colton, M. Cook, & D. Ventura (Eds.), *Proceedings of the 6th international conference on computational creativity, ICCC 2015* (pp. 276–283).

Kassam, K. S., & Mendes, W. B. (2013). The effects of measuring emotion: Physiological reactions to emotional situations depend on whether someone is asking. *PloS One*, *8*(6).

Kaufman, J. C., & Beghetto, R. A. (2009). Beyond big and little: The Four C model of creativity. *Review of General Psychology*, *13*, 1–12.

Krippendorf, K. (1980). *Content analysis: An introduction to its methodology*. Thousand Oaks, CA: Sage Publications.

Lamb, C., Brown, D. G., & Clarke, C. L. (2015). Human competence in creativity evaluation. In H. Toivonen, S. Colton, M. Cook, & D. Ventura (Eds.), *Proceedings of the 6th international conference on computational creativity, ICCC 2015* (pp. 102–109).

Lenat, D. B. (1976). *An artificial intelligence approach to discovery in mathematics as heuristic search* (Memo No. AIM-286). Department of Computer Science, Stanford University. Stanford, CA.

Linson, A., Dobbyn, C., & Laney, R. (2012). Critical issues in evaluating freely improvising interactive music systems. In M. L. Maher, K. Hammond, A. Pease, R. Pérez y Pérez, D. Ventura, & G. Wiggins (Eds.), *Proceedings of the 3rd international conference on computational creativity, ICCC 2012* (pp. 145–149).

Llano, M. T., Colton, S., Hepworth, R., & Gow, J. (2016). Automated fictional ideation via knowledge base manipulation. *Cognitive Computation*, *8*(2), 153–174.

Loebner Prize. (n.d.). http://www.loebner.net/Prizef/loebner-prize.html. accessed May 2017.

Macedo, L., & Cardoso, A. (2001). Creativity and surprise. In G. Wiggins (Ed.), *Proceedings of the AISB 01 symposium on artificial intelligence and creativity in arts and science* (pp. 84–92). Society for the Study of Artificial Intelligence and Simulation of Behaviour.

Maher, M. L., Brady, K., & Fisher, D. H. (2013). Computational models of surprise in evaluating creative design. In M. L. Maher, T. Veale, R. Saunders, & O. Bown (Eds.), *Proceedings of the 4th international conference on computational creativity, ICCC 2013* (pp. 147–151).

Manurung, R., Ritchie, G., Pain, H., Waller, A., O'Mara, D., & Black, R. (2008). The construction of a pun generator for language skills development. *Applied Artificial Intelligence*, *22*(9), 841–869.

McDermott, D. V. (1976). Artificial intelligence meets natural stupidity. *SIGART Bulletin*, *57*, 4–9.

Moffat, D., & Kelly, M. (2006). An investigation into people's bias against computational creativity in music composition. In *Proceedings of 3rd joint workshop on computational creativity, ECAI* (pp. 20–25). Riva del Garda, Italy.

Monteith, K., Martinez, T., & Ventura, D. (2010). Automatic generation of music for inducing emotive response. In D. Ventura, A. Pease, R. Pérez y Pérez, G. Ritchie, & T. Veale (Eds.), *Proceedings of the 1st international conference on computational creativity, ICCC 2010* (pp. 140–149). Department of Informatics Engineering, University of Coimbra.

Mumford, M., & Ventura, D. (2015). The man behind the curtain: Overcoming skepticism about creative computing. In H. Toivonen, S. Colton, M. Cook,

& D. Ventura (Eds.), *Proceedings of the 6th international conference on computational creativity, ICCC 2015* (pp. 1–6).

Neuendorf, K. A. (2002). *The content analysis guidebook*. Thousand Oaks, CA.: Sage Publications.

Norton, D., Heath, D., & Ventura, D. (2010). Establishing appreciation in a creative system. In D. Ventura, A. Pease, R. Pérez y Pérez, G. Ritchie, & T. Veale (Eds.), *Proceedings of the 1st international conference on computational creativity, ICCC 2010* (pp. 26–35). Department of Informatics Engineering, University of Coimbra.

Norton, D., Heath, D., & Ventura, D. (2015). Accounting for bias in the evaluation of creative computational systems: An assessment of DARCI. In H. Toivonen, S. Colton, M. Cook, & D. Ventura (Eds.), *Proceedings of the 6th international conference on computational creativity, ICCC 2015* (pp. 31–38).

Papineni, K., Roukos, S., Ward, T., & Zhu, W.-J. (2002). BLEU: A method for automatic evaluation of machine translation. In *Proceedings of the 40th annual meeting of the association for computational linguistics* (pp. 311–318). Philadelphia.

Pearce, M. T., Meredith, D., & Wiggins, G. A. (2002). Motivations and methodologies for automation of the compositional process. *Musicae Scientae*, *6*(2), 119–147.

Pearce, M. T., & Wiggins, G. A. (2001). Towards a framework for the evaluation of machine compositions. In *Proceedings of the AISB symposium on artificial intelligence and creativity in the arts and sciences* (pp. 22–32). York, UK.

Pearce, M. T., & Wiggins, G. A. (2007). Evaluating cognitive models of musical composition. In G. Wiggins & A. Cardoso (Eds.), *Proceedings of the 4th international joint workshop on computational creativity* (pp. 73–80). London, UK.

Pease, A., & Colton, S. (2011a). Computational creativity theory: Inspirations behind the FACE and the IDEA models. In D. Ventura, P. Gervás, D. Fox Harrell, M. L. Maher, A. Pease, & G. Wiggins (Eds.), *Proceedings of the 2nd international conference on computational creativity, ICCC 2011* (pp. 72–77).

Pease, A., & Colton, S. (2011b). On impact and evaluation in computational creativity: A discussion of the Turing test and an alternative proposal. In *Proceedings of the AISB symposium on AI and philosophy*.

Pease, A., Winterstein, D., & Colton, S. (2001). Evaluating machine creativity. In R. Weber & C. G. von Wangenheim (Eds.), *Case-based reasoning: Papers from the workshop programme at ICCBR 01* (pp. 129–137). Vancouver.

Peinado, F., & Gervás, P. (2006). Evaluation of automatic generation of basic stories. *New Generation Computing*, *24*(3), 289–302.

Pereira, F. C. (2005). *A computational model of creativity* (Doctoral dissertation, Universidade de Coimbra).

Pereira, F. C., Mendes, M., Gervás, P., & Cardoso, A. (2005). Experiments with assessment of creative systems: An application of Ritchie's criteria. In P. Gervás, T. Veale, & A. Pease (Eds.), *Proceedings of the workshop on computational creativity, 19th international joint conference on artificial intelligence*

(pp. 37–44). Technical Report 5-05. Departamento de Sistemas Informáticos y Programación, Universidad Complutense de Madrid.

Pérez y Pérez, R., & Ortiz, O. (2013). A model for evaluating interestingness in a computer-generated plot. In M. L. Maher, T. Veale, R. Saunders, & O. Bown (Eds.), *Proceedings of the 4th international conference on computational creativity, ICCC 2013* (pp. 131–138).

Pérez y Pérez, R., Ortiz, O., Luna, W., Negrete, S., Castellanos, V., Peñalosa, E., & Ávila, R. (2011). A system for evaluating novelty in computer generated narratives. In D. Ventura, P. Gervás, D. Fox Harrell, M. L. Maher, A. Pease, & G. Wiggins (Eds.), *Proceedings of the 2nd international conference on computational creativity, ICCC 2011* (pp. 63–68).

Picard, R. W. (2000). *Affective computing*. Cambridge, Mass: MIT Press.

Puccio, G. J., & Murdock, M. C. (Eds.). (1999). *Creativity assessment: Readings and resources*. Hadley, Mass.: Creative Education Foundation.

Ritchie, G. (2001). Assessing creativity. In *Proceedings of the AISB symposium on artificial intelligence and creativity in arts and science* (pp. 3–11). York, England.

Ritchie, G. (2006). The transformational creativity hypothesis. *New Generation Computing*, *24*, 241–266.

Ritchie, G. (2007). Some empirical criteria for attributing creativity to a computer program. *Minds and Machines*, *17*(1), 67–99.

Ritchie, G. (2008). Uninformed resource creation for humour simulation. In P. Gervás, R. Pérez y Pérez, & T. Veale (Eds.), *Proceedings of the 5th international joint workshop on computational creativity* (IT/2008/2, pp. 147–150). Technical Report. Universidad Complutense de Madrid.

Ritchie, G. (2011). The formal description of computational creativity. Presentation. 1st Autumn School on Computational Creativity, Porvoo, Finland.

Ritchie, G. (2012). A closer look at creativity as search. In M. L. Maher, K. Hammond, A. Pease, R. Pérez y Pérez, D. Ventura, & G. Wiggins (Eds.), *Proceedings of the 3rd international conference on computational creativity, ICCC 2012* (pp. 41–48).

Ryan, R. M., & Deci, E. L. (2000). Intrinsic and extrinsic motivations: Classic definitions and new directions. *Contemporary Educational Psychology*, *25*, 54–67.

Schmidhuber, J. (2010). Formal theory of creativity, fun, and intrinsic motivation (1990-2010). *IEEE Transactions on Autonomous Mental Development*, *2*(3), 230–247.

Shneiderman, B., & Plaisant, C. (2006). Strategies for evaluating information visualization tools: Multi-dimensional in-depth long-term case studies. In *Proceedings of the 2006 AVI workshop on BEyond time and errors: Novel evaluation methods for information visualization* (pp. 1–7). Venice, Italy: ACM.

Takala, T. (2015). Preconceptual creativity. In H. Toivonen, S. Colton, M. Cook, & D. Ventura (Eds.), *Proceedings of the 6th international conference on computational creativity, ICCC 2015* (pp. 252–259).

Tearse, B., Mawhorter, P., Mateas, M., & Wardrip-Fruin, N. (2011). Experimental results from a rational reconstruction of MINSTREL. In D. Ventura, P. Gervás, D. Fox Harrell, M. L. Maher, A. Pease, & G. Wiggins (Eds.), *Proceedings of the 2nd international conference on computational creativity, ICCC 2011* (pp. 54–59).

Tobing, B. C. L., & Manurung, R. (2015). A chart generation system for topical metrical poetry. In H. Toivonen, S. Colton, M. Cook, & D. Ventura (Eds.), *Proceedings of the 6th international conference on computational creativity, ICCC 2015* (pp. 308–314).

Turing, A. (1950). Computing machinery and intelligence. *Mind*, *59*(236), 433–460.

van der Sluis, I., & Mellish, C. (2008). Towards affective natural language generation: Empirical investigations. In C. Mellish (Ed.), *Proceedings of the symposium on affective language in human and machine, AISB 2008* (pp. 9–16). Society for the Study of Artificial Intelligence and Simulation of Behaviour.

van der Velde, F., Wolf, R. A., Schmettow, M., & Nazareth, D. S. (2015). A semantic map for evaluating creativity. In H. Toivonen, S. Colton, M. Cook, & D. Ventura (Eds.), *Proceedings of the 6th international conference on computational creativity, ICCC 2015* (pp. 94–99).

Veale, T. (2014). Coming good and breaking bad: Generating transformative character arcs for use in compelling stories. In S. Colton, D. Ventura, N. Lavrač, & M. Cook (Eds.), *Proceedings of the 5th international conference on computational creativity, ICCC 2014* (pp. 239–246).

Veale, T. (2015). Scoffing at mere generation. Blog entry . http://prosecco-network.eu/blog/scoffing-mere-generation.

Ventura, D. (2008). A reductio ad absurdum experiment in sufficiency for evaluating (computational) creative systems. In P. Gervás, R. Pérez y Pérez, & T. Veale (Eds.), *Proceedings of the 5th international joint workshop on computational creativity* (IT/2008/2, pp. 11–19). Technical Report. Universidad Complutense de Madrid.

Ventura, D. (2016). Mere generation: Essential barometer or dated concept? In F. Pachet, A. Cardoso, V. Corruble, & F. Ghedini (Eds.), *Proceedings of the 7th international conference on computational creativity, ICCC 2016* (pp. 17–24).

Watson, D., Clark, L. A., & Tellegen, A. (1988). Development and validation of brief measures of positive and negative affect: The PANAS scales. *Journal of Personality and Social Psychology*, *54*(6), 1063–1070.

Wiggins, G. (2001). Towards a more precise characterisation of creativity in AI. In R. Weber & C. G. von Wangenheim (Eds.), *Case-based reasoning: Papers from the workshop programme at ICCBR 2001* (pp. 113–120). Vancouver, BC, Canada: Navy Center for Applied Research in Artificial Intelligence.

Wiggins, G. (2003). Categorising creative systems. In *Proceedings of 3rd (IJCAI) workshop on creative systems: Approaches to creativity in artificial intelligence and cognitive science*, Acapulco, Mexico.

Wiggins, G. (2005). Searching for computational creativity. In P. Gervás, T. Veale, & A. Pease (Eds.), *Proceedings of the IJCAI-05 workshop on computational*

creativity (pp. 68–73). Technical Report 5-05. Departamento de Sistemas Informáticos y Programación, Universidad Complutense de Madrid.

Wiggins, G. (2006a). A preliminary framework for description, analysis and comparison of creative systems. *Knowledge-Based Systems*, *19*, 449–458.

Wiggins, G. (2006b). Searching for computational creativity. *New Generation Computing*, *24*(3), 209–222.

Wiggins, G. (2017). A framework for the description, analysis and comparison of creative systems. In T. Veale & F. A. Cardoso (Eds.), *Computational creativity: The philosophy and engineering of autonomously creative systems*. Computational Synthesis and Creative Systems. Springer International Publishing.

Winograd, T. (1972). *Understanding natural language*. New York: Academic Press.

Winston, P. H. (1970). *Learning structural descriptions from examples* (tech. rep. No. MIT/LCS/TR-76). Massachusetts Institute of Technology. Cambridge, Mass.

Zhu, J. (2012). Towards a mixed evaluation approach for computational narrative systems. In M. L. Maher, K. Hammond, A. Pease, R. Pérez y Pérez, D. Ventura, & G. Wiggins (Eds.), *Proceedings of the 3rd international conference on computational creativity, ICCC 2012* (pp. 150–154).

Chapter 9
Expectation-Based Models of Novelty for Evaluating Computational Creativity

Kazjon Grace and Mary Lou Maher

Abstract The most broadly agreed-upon definition of a creative artefact is one that is both novel and valuable. This chapter argues for measuring novelty as the violation of observers' expectations instead of as the objective differences between artefacts. We discuss why models of novelty based on measuring the differences between known and new artefacts are insufficient, and how models based on expectation can address those insufficiencies. We also describe how expectations can influence the generation of new creative artefacts, showing how unexpected discoveries can focus attention and reformulate goals. We base this on cognitive science research that shows this behaviour during creative design in humans. We present formal notation for a simple model of expectation, and give examples from our implementations in the domains of product design and culinary creativity.

9.1 Introduction: Novelty as Violated Expectations

Research in computational creativity has established the importance of the capacity of a creative system to evaluate its output; see chapters of this volume: Jordanous (2019) and Ritchie (2019). While there is strong agreement that evaluating the creativity of an artefact is an integral part of computational creativity, there is significant disagreement as to how this evaluation can be modelled. One of the more widely accepted characterisations of a creative artefact is that it is simultaneously novel (or unexpected, or original) and valuable (or appropriate, or useful) (Csikszentmihalyi & Wolfe, 2014; Grace, Maher, Fisher, & Brady, 2015; Macedo, Cardoso, Reisenzein, Lorini, & Castelfranchi, 2009; Newell, Shaw, & Simon, 1959). In this chapter we argue that novelty cannot be effectively evaluated by measuring the differences

Kazjon Grace
UNC Charlotte, NC, USA. e-mail: k.grace@uncc.edu

Mary Lou Maher
UNC Charlotte, NC, USA. e-mail: m.maher@uncc.edu

T. Veale, F. A. Cardoso (eds.), *Computational Creativity*, Computational Synthesis and Creative Systems, https://doi.org/10.1007/978-3-319-43610-4_9

between artefacts; instead, novelty should consider *observers' expectations*. An unexpected artefact is one that violates an observer's confident expectations built from their prior experiences with artefacts of that kind. This makes unexpected artefacts novel relative to that individual's experiences, rather than novel when compared objectively with all existing artefacts. Using expectation-based models of novelty (in conjunction with models of value) shifts the focus from artefact to observer, includes a temporal component in the creative experience, and better captures the role of perception in creativity (Grace & Maher, 2015a, 2015b, 2016; Grace et al., 2015).

While we focus on the novelty-and-value definition in this chapter, there is significant diversity in the literature on evaluating creativity (Jordanous, 2011). Pease and Colton (2011) focus on whether a creative computer-generated artefact is indistinguishable from human creative output; Boden (2003) and Wiggins (2006b) claim an artefact is creative when it causes a transformation of the way society thinks about the creative domain; Koestler (1964) claims an artefact is creative when it causes a novel blending of two formerly disparate conceptual spaces; and Csikszentmihalyi (1988) and Amabile (1996) argue that creativity can only be evaluated by society, with the latter allowing a panel of expert (human) judges to stand in society's stead. There is not room here to even mention all possible perspectives on the notion of creativity – one review lists over 50 definitions in cognitive science alone (Taylor, 1988). In this chapter we put forward expectation-based models as a critical component of measuring novelty. This approach supersedes the difference-based models of novelty used in many novelty-and-value measures of creativity.

Modelling unexpectedness reimagines what makes a creative artefact different, focusing not on objective comparisons but on subjective perceptions. This concept has been explored under the term 'surprise' in past literature (Grace et al., 2015; Macedo & Cardoso, 2001; Ortony & Partridge, 1987), a term that also has been used to refer to an observer's *response* to a novel or unexpected object, rather than an attribute of the object itself (Wiggins, 2019). In this chapter we define a novel object as one that violates an observer's confident prior expectations about its structure or behaviour, and surprise as the observer's response to that unexpectedness. Novel artefacts are unexpected by observers, and observers are surprised by novel artefacts. This distinction allows us to explore unexpectedness separately from how a creative system reacts to it (see Section 9.3).

This 'eye of the beholder' approach framing is compatible with formulations of creativity that focus not on artefacts but on their artificers and the society and cultures they inhabit (Csikszentmihalyi, 1988). It should be noted that no assumptions are made about the nature of the observer – they may or may not be the artefact's creator, a participant in the domain or a human. The 'observer' may also be broadly construed as society as a whole, representing shared cultural expectations and knowledge. This allows us to distinguish between the p-creativity experienced by an individual observer when their expectations are violated by an artefact they find valuable, and the h-creativity of a 'societal' observer (one that models the perspective of an entire society) experiencing the same, to use the distinction in Boden (2003). For simplicity, we use the term 'observer' throughout this chapter, whether that represents

an individual agent's own perspective or a model reflecting the aggregate perspective of a society.

The 'value' component of the novelty-and-value duality should be interpreted broadly, encompassing artistic, conceptual, personal and social value in addition to money. In fact, additional framing is provided in the earliest reference to the definition of which the authors are aware: a creative artefact has 'novelty and value for the thinker or for [his or her] culture' (Newell et al., 1959). That clarification parallels Boden's distinction between p-creativity for an individual ('the thinker') and h-creativity for society as a whole ('the culture'). Newell, Shaw and Simon are clearly stating that both novelty and value must be placed in an appropriate context, and treated as appropriately subjective and multifaceted. It is clear that a naive monetary metric for value is insufficient: creative artefacts should not be judged solely by their market value. We argue that a naive distance-based measure of novelty that compares artefacts directly is similarly insufficient.

Novelty-as-unexpectedness implies that the observer has an internal model of artefacts in the creative domain. That model is acquired through past observations, which in computational creativity typically involves a machine learning process. Unlike distance-based measures of novelty, this approach can capture relationships between the features of an artefact, allowing innovative combinations of existing artefacts to be considered creative even if they lie within previously explored regions of the conceptual space. Expectation can also incorporate knowledge that is extrinsic to an artefact, such as its maker, its genre, its critical reception or its time of release. In this way an artefact can be unexpected relative to domain trends, or relative to its categorisation.

This chapter serves as an introduction to the use of expectation-based models in computational creativity. It provides background on why unexpectedness is a useful component of creativity evaluation, and how it compares with other approaches. It also provides a simple theoretical framework for describing expectation-based evaluation, as well as examples of implementations from our own work and that of others. The latter portion of the chapter describes how unexpectedness can be used as part of a model of curiosity that affects the process a creative system uses to generate artefacts. Surprise-triggered specific curiosity provides a model for a creative system to iteratively reinterpret and reformulate a problem while solving it, a behaviour which cognitive studies suggest is critical to human creativity (Schön, 1983).

9.2 Expectation-Based Novelty for Evaluating Creative Artefacts

Unexpectedness has been described as the violation of a confident belief (Ortony & Partridge, 1987), an automatic reaction to a mismatch (Lorini & Castelfranchi, 2007), an input–expectation discrepancy (Partridge, 1985) and an emotional response to novelty (Wiggins, 2006a). Some cognitive characterisations of surprise separate active expectations (those explicitly formed prior to the surprising event) from passive ones (post-hoc expectations arising from previous experience only after the unlikely

perceptual datum is encountered) (Lorini & Castelfranchi, 2007; Macedo & Cardoso, 2001; Ortony & Partridge, 1987). Others focus on distinguishing low-level sensory from higher-level symbolic expectations (Kahneman & Tversky, 1982), and yet more on expectation failures due to ignorance or unreliable evidence as opposed to the genuinely surprising failure of confident expectations (Ortony & Partridge, 1987). Additionally, a variety of computational frameworks for formalising unexpectedness and surprise have been proposed (Grace, Maher, Fisher, & Brady, 2014; Horvitz, Apacible, Sarin, & Liao, 2012; Macedo & Cardoso, 2001; Macedo et al., 2009; Maher & Fisher, 2012).

These observer-centric views of novelty permit a much richer notion of what makes an artefact different: it might relate to the subversion of established power structures (Florida, 2012), the destruction of established processes (Schumpeter, 1942), or the transgression of established rules (Dudek, 1993; Strzalecki, 2000). These kinds of cultural impacts go beyond a more simple notion of originality, that is, an artefact can be different from existing artefacts but that difference is expected. For example, we have expectations about what the next model of mobile phones will be, and the difference alone does not make the next model creative. A creative mobile phone is one whose differences are unexpected and valuable, not simply novel and valuable. This observer-based perspective of evaluating expectation and its counterpart, unexpectedness, is the basis for a computational model that can evaluate creativity.

9.2.1 A Formal Model of Expectation-Based Novelty

For the purposes of this introductory chapter, we will present a framework that is compatible with the majority of the prior work and easily portable to new creative domains. We measure unexpectedness probabilistically, as the complement of the observer's a priori expected likelihood of making an observation. The less likely an observer believes an artefact to be, the more surprising it is to that observer when that artefact occurs. Each expectation is associated with a confidence, which is an estimate of the reliability of the expected likelihood given the observers' experiences related to the new artefact.

Confidence serves to distinguish unexpectedness resulting from the violation of a confident belief (which is associated with creativity) from unexpectedness resulting from ignorance (which is not). This confidence can be expressed using methods like a margin of error or a credible interval. Some methods for modelling expectation (such as those based on Bayesian statistics) incorporate confidence directly into their likelihood estimates (Pearl, 1986), rendering a separate confidence measure unnecessary, but we include it so as not to exclude other approaches.

Formally, consider an artefact $a \in A$, where A is the set of all possible artefacts. Each artefact consists of a set of features $f \in F$, where F denotes the space of all possible features. An artefact is perceived by the creative system (our "observer") to have a set of features, which we denote $a = \{f_1, f_2, \ldots, f_n\}$. This allows for any

discretely valued artefact attributes to be represented by a set of possible features, and for any continuously valued attributes to be represented by a similar infinite set. Let us leave aside, for now, the question of how that infinite set would work in practice. The goal here is to describe unexpectedness in a generally applicable way, not an immediately implementable one. Assume that our creative system has observed some subset of possible artefacts $K \subset A$, the 'known set' from which expectations are formed.

We can now develop a model of an individual unexpected observation. We describe this as making a prediction about a 'predicted' set of features $d \subset a$ based on a 'predictor' set of features $r \subset a$. The creative system has some expectation about what should make up the rest of an artefact that contains r, from which the likelihood of the features in d can be inferred. We define an expectation function x:

$$x : (r, d, K) \rightarrow (p(f_d), c(f_d), \forall f_d \in d)$$

which takes as input the predictor features r, the predicted features d and the known artefacts K. Unexpectedness produces a probability p and a confidence c for each feature in d. Put another way: expectation is a process that uses known artefacts (K) to make a prediction about how likely it is that some features (d) will be observed alongside if some other features (r) are also present. In the special case where r is empty, the system produces an expectation for the likelihood of the features in d that is not conditioned on observing any other features. This can be considered equivalent to non-expectation-driven models of novelty. An unexpectedness function is then used to measure the output of expectation:

$$u : (1 - p(f_d), c(f_d), K) \rightarrow [0, 1]$$

which converts the likelihood and confidence values for each feature into a scalar value. There are several possible choices for this function, depending on how likelihood and confidence are modelled: in some cases they can be simply multiplied together to give a confidence-weighted likelihood, as in Grace et al. (2014). In other cases the unexpectedness measure can be calculated from the effect the observation has on the model's expectation function, which is why the set of known artefacts is included as input. This second approach to calculating unexpectedness is based on the *transformational creativity* approach (Boden, 2003; Grace et al., 2015; Wiggins, 2006b). A consequence of this approach is that a large number of possible unexpected observations can be made about any artefact, as any combination of its features can be used in both the predictor and the predicted set. This creates the question of how to aggregate these values to rate an artefact as a whole, for which both the maximal individual unexpectedness (Grace et al., 2014) and the average over all unexpectedness values (Macedo & Cardoso, 2001) have been proposed.

The key questions when implementing this framework are what should constitute the features to be expected, and through what algorithms should expectations about them be formed? The former is a familiar question to all researchers in AI, as proper representation is and has always been crucial to any intelligent system. The second

question is one that can seem equally overwhelming, but we have developed several recommendations:

1. Supervised machine learning algorithms can be used to learn to predict one feature based on one or more other features. If the predicted feature is categorical, consider classification algorithms. If it is ordinal or numerical, consider regression algorithms.
2. Probabilistic graphical models (Koller & Friedman, 2009), including Markov models, are a natural fit for representing conditional probabilities, but can be less easy to apply out-of-the-box than supervised learning algorithms as they often require making more assumptions when constructing the model.
3. Deep autoencoders (Bengio, Courville, & Vincent, 2013) combine elements of both feature selection and expectation modelling. They aim to learn a compressed representation that can "reconstruct" each object, which necessitates learning what to expect about each feature when given values for others.

These are, of course, only guidelines, and the development of suitable expectation models for different creative domains is an active area of research.

9.2.2 Related Approaches to Creativity Evaluation

Expectation-based models overcome the insufficiency of difference-based novelty by replacing their distance measures with a probabilistic model of likelihood. Another approach to the same problem is to measure novelty in a *conceptual space*, rather than in the space of possible representations. Conceptual spaces (Boden, 2003; Gärdenfors, 2004) are characterised by dimensions that are meaningful to an observer for a category of artefact, and are typically thought of as being learnt through experience. Saunders and Gero (2001b) show one path to operationalising this, using unsupervised learning to emergently define the underlying structure of the domain's conceptual space through observation. These approaches improve upon novelty measures calculated in the representational space of artefacts by providing experientially grounded subjectivity. Expectation and conceptual spaces are not incompatible approaches, as unexpectedness could be evaluated using emergent conceptual representations.

Maher and Fisher (2012) propose a tripartite evaluation model in which novelty, value and surprise are all essential components of creativity. They view novelty and surprise as distinct, with novelty being based on distance-based measures in a conceptual space (created by clustering), and surprise being based on temporal trends within the dataset. In this chapter we argue that expectation-based models include and go beyond expectations that are related to time by modelling expectations formed about any combination of features or concepts, thereby superseding novelty as difference. As our notion of unexpectedness can also encompass cluster-based distance measures of novelty (through non-conditional expectations where $r = \emptyset$)

we can model surprise as a response to expectation-based novelty, rather than as a separate factor in creativity (Grace & Maher, 2014).

Boden (2003) proposed the notion of transformational creativity, later operationalised in Wiggins (2006b), as an alternative means for recognising the creativity of an artefact. Transformational creativity – the degree to which an artefact changes the creative domain to which it belongs – captures the idea that creativity disrupts and revolutionises. This is suggested by Boden to be a more significant form of creativity than the combination of 'mere' novelty and value. Grace and Maher (2014) argue that transformational creativity occurs as a collective societal reaction to the observation of an unexpected design – in other words, artefacts that cause domain transformation will also be surprising. Grace et al. (2015) developed a model of surprise based on the impact an artefact had on an agent's knowledge, which could be used as a building block for operationalising transformation at the societal level. Our model of expectation-based novelty operationalises the concept of transformational creativity, where the notions of disruption and revolution in Wiggins' and Boden's accounts are related to the magnitude of surprise and unexpectedness. We claim that experiences of creativity other than transformational are not 'mere' novelty where novelty is simply difference, but are unexpected even if not revolutionary.

9.2.3 Implementing Expectation-Based Novelty

We have applied the creativity evaluation framework described above to the product design domain, specifically a database of mobile devices from 1985 to 2014 (Grace & Maher, 2015b; Grace et al., 2014, 2015). We developed a series of expectation models, each predicting one attribute of a device based on a single other attribute. Each model had a single feature in d and a single feature in r, and there were 132 expectation models consisting of all pairwise comparisons between 12 attributes. We used a confidence measure based on predictive error to ensure that models with insufficient predictive value did not influence unexpectedness. The system rated devices like the LG KC-1, which had a much faster CPU than devices released around the same time (see Fig. 1), as highly unexpected. Other discoveries include the original Apple iPad, which combined the large physical size of older devices with technical specifications that outperformed modern phones.

The system S-EUNE (Macedo & Cardoso, 2001) is another implementation of this kind of expectation-based creativity evaluation. S-EUNE is an artificial agent that explores uncertain and unknown environments using surprise as an emotional driver. It partially observes objects such as buildings from a distance (forming its predictor set r), and uses that information to build expectations about what it will see once it gets closer (its predicted set d). The agent then either confirms its expectations or becomes surprised when they are violated. In S-EUNE the expectation is made before the true value of the predicted set is known, a form of active expectation (Ortony & Partridge, 1987).

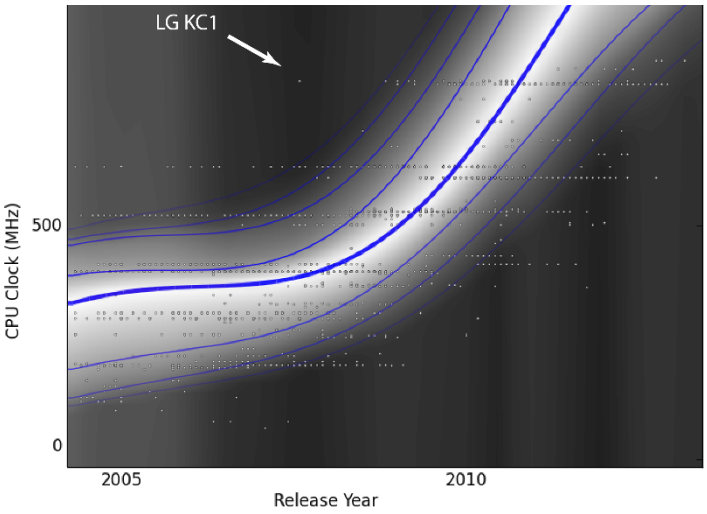


Fig. 9.1 Expectations about a mobile device's CPU speed given its year of release. Devices are marked by dots, the blue lines indicate contours of the expected distribution (the thickest line is the median) and the background shading indicates the confidence-weighted expected likelihood of observing a new design. A highly unexpected mobile device, the LG KC-1, is indicated.

IDyOT, or 'Information Dynamics Of Thinking' (Wiggins & Forth, 2015), is a cognitive architecture for computational creativity in sequential domains such as music or language. IDyOT uses expectation-based evaluation to predict the next symbol in the sequence it is perceiving. It is based on the hypothesis that spontaneous (as opposed to deliberate) creativity can be explained by hierarchical predictive models built from observations of prior input. The model, when given the task of segmenting sequences of phonemes into words, finds the final word unexpected in phrases like 'the horse raced past the barn fell', as after 'barn' it anticipates that the phrase is complete. Competing predictors emerge as a sequence is being parsed, each offering a distribution of expected next symbols based on their interpretation of the sequence in progress. IDyOT is an expectation-based model in which artefacts are a univariate sequence (as in music and language, among other domains), predicting the next symbol in the sequence (d) based on those that preceded it (r).

These three systems demonstrate the flexibility of the expectation-based approach: it can be applied to any creative domain for which a predictive model can be constructed. Supervised machine learning models can be applied to predict features based on other features. Classifiers can be used to predict categorical features, and regression models to predict continuous ones. Unsupervised models such as deep autoencoders (Rezende, Mohamed, & Wierstra, 2014; Vincent, Larochelle, Lajoie, Bengio, & Manzagol, 2010) can be used to learn self-completing representations and make predictions about artefacts as a whole. Techniques from natural language processing such as topic modelling (Blei, 2012) or neural network based word embedding (Mikolov, Chen, Corrado, & Dean, 2013) can be used on unstructured data.

The expectation-based approach does not rely on any particular implementation, but instead offers a general framework for data-driven creativity evaluation.

9.3 How Expectation-Based Novelty Affects the Generation of Creative Artefacts

One way creative systems can exhibit greater autonomy is by pursuing artefacts or concepts that interest them (Saunders, 2012), and unexpected artefacts are interesting to their creators. This behaviour is referred to as *specific curiosity* (Grace & Maher, 2015a), and it can lead to a series of related artefacts embodying a particular combination of concepts on which the creator has fixated – a very human-like behaviour. A generative system with novelty-based evaluation will produce artefacts that are unexpected but unrelated to each other: novelty-based evaluation does not care *why* an artefact is novel, only that it is. Specific curiosity allows systems to focus their attention on a particular surprising combination, and pursue it by generating artefacts until that combination is no longer unexpected (i.e. it is 'understood'). Human creators appear to do this regularly, pursuing a particular concept that interests them both while generating a single artefact and across entire creative careers. Expectation-based models can be used to build systems that exhibit this behaviour (Grace & Maher, 2015a, 2015b; Merrick & Maher, 2009; Saunders & Gero, 2001a).

The term 'specific curiosity' comes from Berlyne (1966), who distinguishes curiosity along two axes: perceptual vs epistemic and diversive vs specific. Both are critical to computational creativity. Perceptual curiosity is the drive towards novel sensory stimuli, and has been observed in both animals and humans. Epistemic curiosity is the uniquely human drive to acquire new knowledge. This conceptual form of curiosity has been modelled by systems that learn a conceptual space and measure novelty within it, rather than measuring novelty using objective representations (Saunders & Gero, 2001a). Diversive curiosity, on which most computational models of curiosity have focused, is the search for new information of any kind. Specific curiosity is the search for observations that explain or elaborate on a particular goal concept. For example, a musician may be driven to recreate a particular "sound" that emerged from improvisation, or a chef may be obsessed with exploring a newly discovered exotic ingredient.

Specific curiosity is related to the notions of reflection, problem framing and goal formulation (Cross, 2004; Getzels & Csikszentmihalyi, 1976). The exact nature of the interactions between those notions and the search for solutions is an active area of design cognition research (Schön, 1983). Problem framing can be triggered by unexpected discoveries, i.e. the observation of a design of low prior likelihood (Suwa, Gero, & Purcell, 1999). Human designers have the ability to surprise themselves by creating intermediate external representations (e.g. sketches) while designing, and these self-surprises lead to greater creativity (Suwa et al., 1999). Discovering unexpected things in one's own emerging creative artefacts can lead a creator to

pursue a new approach, which in turn can lead to yet more unexpected discoveries (Suwa, Gero, & Purcell, 2000).

The challenge for operationalising specific curiosity is how it is triggered: when and why should a creative system become specifically curious? This is related to the broader issue of creative autonomy (Jennings, 2010; Saunders, 2012). For a more in-depth discussion of the role of specific curiosity, unexpectedness and creative autonomy see Grace and Maher (2015a, 2015b).

9.3.1 Implementing Expectation-Based Generation

We present here a computational framework in which the surprise caused by observing an unexpected artefact leads the creative system to reformulate its goals. We do this by adopting a metacognitive perspective, in which generation and evaluation occur at the cognitive level while surprise-triggered reformulation occurs at the metacognitive (i.e. reasoning about reasoning) level. This parallels AI models of metacognition, in which the detection of an anomalous stimulus is used as a trigger for the reformulation of goals (Cox & Raja, 2011). Figure 9.2 shows the structure of the framework.

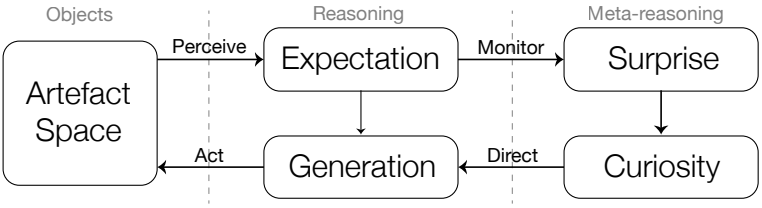


Fig. 9.2 Specific curiosity's metacognitive role in creativity (after Grace & Maher, 2015b), showing levels of reasoning as per Cox and Raja (2011).

At the object level is the space of artefacts, both known and hypothetical. At the reasoning level, the system forms expectations about artefacts it observes (using something akin to the framework described in Section 9.2) and generates new artefacts by whatever means. At the meta-reasoning level, the system exhibits surprise when it encounters a highly unexpected artefact, leading to curiosity: the drive to explore a stimulus, and other stimuli similar to it, until its behaviour can be explained. Curiosity directs the generation of new artefacts by changing the system's goals to favour artefacts that are unexpected *in a similar way* to those of the triggering surprise (Grace & Maher, 2015a, 2015b). During this curiosity-driven period of generation any artefacts that are unexpected in a totally different way from the triggering stimulus are not considered to be novel.

We are developing an implementation of specific curiosity in the domain of recipe generation called 'Q-chef (short for 'curious chef'). Our system responds to

unexpected recipes by trying to generate recipes that contain similar unexpected ingredient combinations (Grace & Maher, 2016). Trained on a library of 130,000 recipes scraped from the web, Q-chef responded to a surprising chocolate–bacon cupcake recipe (in which bacon was unexpected given the presence of cocoa, sugar and butter) by generating the three recipes in Table 9.1. In terms of the model in Fig. 9.2, Table 9.1 shows three examples of an *Expectation* → *Surprise* → *Curiosity* → *Generation* cycle. These three examples differ in how the similarity between surprises is calculated for the purposes of curiosity influencing generation. Q-chef uses a probabilistic deep neural network called a Variational Autoencoder (Kingma & Welling, 2013; Rezende et al., 2014) to capture expectations about ingredient pairings. The system serves as a proof of concept for how surprise and curiosity can drive generative processes.

Table 9.1 Recipes generated in response to a chocolate-bacon cupcake recipe in which bacon was unexpected given the presence of *cocoa*, *butter* and *sugar*. In each recipe, the most surprising predictor *r* is italicised and the predicted set *d* is bolded. The interpretations were added by the authors as Q-chef currently only models recipes as sets of ingredients.

#	*Ingredients*	*Interpretation*
1.	Rye flour, flour, coffee, water, salt, caraway seeds, yeast, molasses, *cocoa*, *sugar*, *butter* and **bacon**.	Rye bread with coffee/bacon.
2.	Pasta, *pecans*, cheese, eggs, parmesan, butter, *dill*, *vodka*, white wine, salt, black pepper and **bacon**.	Rich and cheesy pecan–bacon pasta.
3.	**Sweet potatoes**, coconut, eggs, flour, vanilla, anise, cinnamon, *cocoa*, *butter* and *sugar*.	Coconut–chocolate sweet potato casserole.

The three recipes in Table 9.1 showcase different approaches to calculating the similarity between what makes recipes unexpected. To produce recipe 1, the system generated recipes with an identical unexpected combination to that in the bacon cupcakes, resulting in something that resembles a flavoured rye bread. For recipe 2, the system was searching for recipes with the same unexpected ingredient in a different context (i.e. a different set of predictors), leading to pasta with a cheese, bacon and vodka sauce. For recipe 3, the system was searching for recipes with the same predictors, but with a different unexpected ingredient, leading to a sweet potato dessert. Further details on these experiments are available in Grace and Maher (2016).

The primary requirement for implementing specific curiosity in a system that already uses unexpectedness-based novelty is a measure of the similarity between two surprising artefacts. For Q-chef we are investigating the use of shared flavour compounds as a measure of ingredient–ingredient similarity (Ahn, Ahnert, Bagrow, & Barabási, 2011), which could be used to relate the ingredients in the predictor and predicted sets of surprising combinations. For example, finding bacon surprising in the presence of butter, flour and sugar would be considered similar to finding salami surprising in the presence of whipped cream, breadcrumbs and maple syrup. Appropriate measures of the similarity between surprises require domain knowledge, and will drive specific curiosity towards domain-appropriate 'interesting' artefacts.

9.4 Discussion

This chapter has focused on the role of expectation in evaluating and generating creative artefacts. We have argued for unexpectedness as a more effective measure of novelty than difference, as it captures the subjectivity of observers' perceptions. Expectation-based models are also better equipped to explain the relationships between artefact features. We presented a simple framework for operationalising expectation, some examples that apply that framework and some ideas on how to apply the approach in other domains.

We have also connected unexpectedness to curiosity, and thereby described how expectation-based models can be used to direct the generation of new creative artefacts. Inspired by results from cognitive science, we argue for models of iterative, parallel interaction between generation and problem-framing based on surprise. We have detailed some of our preliminary results in this area, and discussed how the approach could be implemented in other creative domains.

Expectation-based models, at their core, capture the experience of encountering a creative artefact. As such, they are fundamentally dependent upon our ability to model that perceptual experience computationally. In order to generate artefacts that are considered surprising by humans, a creative system's models of expectation must be able to reflect the expectations of humans as closely as possible – a challenging task. We should not judge creative systems solely by their ability to create artefacts that humans find creative, as artificial creative systems unconcerned with mimicking human surprise may generate surprising artefacts that a human would never have generated. The diverse perspectives offered by such creative systems should be considered a strength.

As artificial intelligence and machine learning progress, we will be able to construct more conceptually complex models of expectation – human-like or otherwise. It will, of course, be some time before an artificial system is able to capture an appreciable fraction of the rich network of concepts a human experiences when encountering a creative work. The power of expectation-based creativity evaluation is that it will grow more accurate as that day approaches.

References

Ahn, Y.-Y., Ahnert, S. E., Bagrow, J. P., & Barabási, A.-L. (2011). Flavor network and the principles of food pairing. *Scientific Reports*, *1*.

Amabile, T. (1996). *Creativity in Context*. Westview Press.

Bengio, Y., Courville, A., & Vincent, P. (2013). Representation learning: A review and new perspectives. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, *35*(8), 1798–1828.

Berlyne, D. E. (1966). Curiosity and exploration. *Science*, *153*(3731), 25–33.

Blei, D. M. (2012). Probabilistic topic models. *Communications of the ACM*, *55*(4), 77–84.

Boden, M. A. (2003). *The Creative Mind: Myths and Mechanisms*. Routledge.

Cox, M. T., & Raja, A. (2011). *Metareasoning: Thinking about Thinking*. MIT Press.

Cross, N. (2004). Expertise in design: An overview. *Design Studies*, *25*(5), 427–441.

Csikszentmihalyi, M. (1988). *Society, Culture, and Person: A Systems View of Creativity*. Cambridge University Press.

Csikszentmihalyi, M., & Wolfe, R. (2014). New conceptions and research approaches to creativity: Implications of a systems perspective for creativity in education. In *The Systems Model of Creativity* (pp. 161–184). Springer.

Dudek, S. Z. (1993). The morality of 20th-century transgressive art. *Creativity Research Journal*, *6*(1-2), 145–152.

Florida, R. L. (2012). *The Rise of the Creative Class: Revisited*. Basic Books.

Gärdenfors, P. (2004). *Conceptual Spaces: The Geometry of Thought*. MIT Press.

Getzels, J. W., & Csikszentmihalyi, M. (1976). *The Creative Vision: A Longitudinal Study of Problem Finding in Art*. Wiley New York.

Grace, K., & Maher, M. L. (2014). What to expect when you're expecting: The role of unexpectedness in computationally evaluating creativity. In *Proceedings of the 4th International Conference on Computational Creativity*.

Grace, K., & Maher, M. L. (2015a). Specific curiosity as a cause and consequence of transformational creativity. In *Proceedings of the 6th International Conference on Computational Creativity* (p. 260).

Grace, K., & Maher, M. L. (2015b). Surprise and reformulation as meta-cognitive processes in creative design. In *Proceedings of the 3rd Annual Conference on Advances in Cognitive Systems (ACS)*.

Grace, K., & Maher, M. L. (2016). Surprise-triggered reformulation of design goals. In *Proceedings of the 30th AAAI Conference on Artificial Intelligence* (pp. 3726–3732). AAAI Press.

Grace, K., Maher, M. L., Fisher, D., & Brady, K. (2014). Data-intensive evaluation of design creativity using novelty, value, and surprise. *International Journal of Design Creativity and Innovation*, *3*(3–4), 125–147.

Grace, K., Maher, M. L., Fisher, D., & Brady, K. (2015). Modeling expectation for evaluating surprise in design creativity. In J. Gero (Ed.), *Design Computing and Cognition 14* (pp. 189–206). Springer.

Horvitz, E. J., Apacible, J., Sarin, R., & Liao, L. (2012). Prediction, expectation, and surprise: Methods, designs, and study of a deployed traffic forecasting service. *arXiv preprint arXiv:1207.1352*.

Jennings, K. E. (2010). Developing creativity: Artificial barriers in artificial intelligence. *Minds and Machines*, *20*(4), 489–501.

Jordanous, A. (2011). Evaluating evaluation: Assessing progress in computational creativity research. In *Proceedings of the Second International Conference on Computational Creativity (ICCC-11). Mexico City, Mexico* (pp. 102–107).

Jordanous, A. (2019). Evaluating evaluation: Assessing progress and practices in computational creativity research. In T. Veale & F. A. Cardoso (Eds.), *Computational creativity: The philosophy and engineering of autonomously creative systems* (pp. 209–234). Springer.

Kahneman, D., & Tversky, A. (1982). Variants of uncertainty. *Cognition*, *11*(2), 143–157.

Kingma, D. P., & Welling, M. (2013). Auto-encoding variational bayes. *arXiv preprint arXiv:1312.6114*.

Koestler, A. (1964). *The Act of Creation*. New York: Macmillan.

Koller, D., & Friedman, N. (2009). *Probabilistic graphical models: Principles and techniques*. MIT Press.

Lorini, E., & Castelfranchi, C. (2007). The cognitive structure of surprise: Looking for basic principles. *Topoi*, *26*(1), 133–149.

Macedo, L., & Cardoso, F. A. (2001). Modeling forms of surprise in an artificial agent. In *Proceedings of the 23rd annual conference of the cognitive science society*, Edinburgh, UK.

Macedo, L., Cardoso, F. A., Reisenzein, R., Lorini, E., & Castelfranchi, C. (2009). Artificial surprise. In J. Vallverdu & D. Casacuberta (Eds.), *Handbook of Research on Synthetic Emotions and Sociable Robotics: New Applications in Affective Computing and Artificial Intelligence* (pp. 267–291). Hershey: Pennsylvania: IGI Global.

Maher, M. L., & Fisher, D. H. (2012). Using AI to evaluate creative designs. In *2nd International Conference on Design Creativity (ICDC)* (pp. 17–19).

Merrick, K., & Maher, M. L. (2009). Motivated Reinforcement Learing. Springer-Verlag, Berlin.

Mikolov, T., Chen, K., Corrado, G., & Dean, J. (2013). Efficient estimation of word representations in vector space. *arXiv preprint arXiv:1301.3781*.

Newell, A., Shaw, J., & Simon, H. A. (1959). *The Processes of Creative Thinking*. Rand Corporation.

Ortony, A., & Partridge, D. (1987). Surprisingness and expectation failure: what s the difference? In *Proceedings of the 10th international joint conference on artificial intelligence* (pp. 106–108). Milan, Italy: Morgan Kaufmann.

Partridge, D. (1985). *Input-Expectation Discrepancy Reduction: A Ubiquitous Mechanism*. Computing Research Laboratory, New Mexico State University.

Pearl, J. (1986). Fusion, propagation, and structuring in belief networks. *Artificial intelligence*, *29*(3), 241–288.

Pease, A., & Colton, S. (2011). On impact and evaluation in computational creativity: A discussion of the turing test and an alternative proposal. In *Proceedings of the AISB symposium on AI and Philosophy*.

Rezende, D. J., Mohamed, S., & Wierstra, D. (2014). Stochastic backpropagation and approximate inference in deep generative models. *arXiv preprint arXiv:1401.4082*.

Ritchie, G. (2019). The evaluation of creative systems. In T. Veale & F. A. Cardoso (Eds.), *Computational creativity: The philosophy and engineering of autonomously creative systems* (pp. 157–192). Springer.

Saunders, R. (2012). Towards autonomous creative systems: A computational approach. *Cognitive Computation*, *4*(3), 216–225.

Saunders, R., & Gero, J. S. (2001a). Artificial creativity: A synthetic approach to the study of creative behaviour. *Computational and Cognitive Models of*

Creative Design V, Key Centre of Design Computing and Cognition, University of Sydney, Sydney, 113–139.

Saunders, R., & Gero, J. S. (2001b). The digital clockwork muse: A computational model of aesthetic evolution. In *Proceedings of the AISB* (Vol. 1, pp. 12–21).

Schön, D. A. (1983). *The Reflective Practitioner: How Professionals Think in Action*. Basic Books.

Schumpeter, J. (1942). *Capitalism, Socialism and Democracy*. Harper & Brothers.

Strzalecki, A. (2000). Creativity in design: General model and its verification. *Technological Forecasting and Social Change*, *64*(2), 241–260.

Suwa, M., Gero, J., & Purcell, T. (2000). Unexpected discoveries and S-invention of design requirements: Important vehicles for a design process. *Design Studies*, *21*(6), 539–567.

Suwa, M., Gero, J., & Purcell, T. (1999). Unexpected discoveries and S-inventions of design requirements: A key to creative designs. *Computational Models of Creative Design IV, Key Centre of Design Computing and Cognition, University of Sydney, Sydney, Australia*, 297–320.

Taylor, C. W. (1988). Various approaches to and definitions of creativity. In R. J. Sternberg (Ed.), *The Nature of Creativity* (pp. 99–121). Cambridge, UK: Cambridge University Press.

Vincent, P., Larochelle, H., Lajoie, I., Bengio, Y., & Manzagol, P.-A. (2010). Stacked denoising autoencoders: Learning useful representations in a deep network with a local denoising criterion. *The Journal of Machine Learning Research*, *11*, 3371–3408.

Wiggins, G. A. (2006a). A preliminary framework for description, analysis and comparison of creative systems. *Knowledge-Based Systems*, *19*(7), 449–458.

Wiggins, G. A. (2006b). Searching for computational creativity. *New Generation Computing*, *24*(3), 209–222.

Wiggins, G. A. (2019). A framework for the description, analysis and comparison of creative systems. In T. Veale & F. A. Cardoso (Eds.), *Computational creativity: The philosophy and engineering of autonomously creative systems* (pp. 21–48). Springer.

Wiggins, G. A., & Forth, J. (2015). IDyOT: A computational theory of creativity as everyday reasoning from learned information. In T. R. Besold, M. Schorlemmer, & A. Smaill (Eds.), *Computational Creativity Research: Towards Creative Machines* (pp. 127–148). Springer.

Chapter 10
Evaluating Evaluation: Assessing Progress and Practices in Computational Creativity Research

Anna Jordanous

Abstract Computational creativity research has produced many computational systems that are described as 'creative'. Historically, these 'creative systems' have not received much in terms of evaluation of the actual creativity of the systems, although this has recently attracted more attention as a research perspective. As a scientific research community, computational creativity researchers can benefit from more systematic/standardised approaches to evaluation of the creativity of our systems, to help us progress in understanding creativity and modelling it computationally. A methodology for creativity evaluation should accommodate different manifestations of creativity but also requires a clear, definitive statement of the tests used for evaluation. Here a historical perspective is given on how computational creativity researchers have evaluated (or not evaluated) the creativity of their systems, considering contextual reasons behind this. Different evaluation approaches and frameworks are currently available, though it is not yet clear which (if any) of several recently proposed methods are emerging as the preferred options to use. The Standardised Procedure for Evaluating Creative Systems (SPECS) forms an overarching set of guidelines for how to tackle evaluation of creative systems and can incorporate recent proposals for creativity evaluation. To help decide which evaluation method is best to use, this chapter concludes by exploring five meta-evaluation criteria devised from cross-disciplinary research into good evaluative practice. Together, these considerations help us explore best practice in computational creativity evaluation, helping us develop the tools we have available to us as computational creativity researchers.

Anna Jordanous
School of Computing, University of Kent, Chatham Maritime, Medway, Kent, UK
e-mail: a.k.jordanous@kent.ac.uk

T. Veale, F. A. Cardoso (eds.), *Computational Creativity*, Computational Synthesis and Creative Systems, https://doi.org/10.1007/978-3-319-43610-4_10

10.1 Introduction

Computational creativity research has produced many computational systems that are described as 'creative'.

We have an intuitive but tacit understanding of the concept of creativity that we can access introspectively (Jordanous, 2012b; Kaufman, 2009). For comparative purposes and methodical, transparent evaluation, this intangible understanding is not sufficient to help us identify and learn from our successes and failures in computational creativity research.

To solve the problem of how to evaluate the *creativity* of computational systems, various evaluation methodologies or strategies have been offered including the tests offered by Pease, Winterstein, and Colton (2001), Ritchie's empirical criteria (Ritchie, 2007, 2019), the creative tripod model (Colton, 2008), the FACE model (Colton, Charnley, & Pease, 2011; Colton, Pease, Corneli, Cook, & Llano, 2014; Pease & Colton, 2011) and, as a more general set of guidelines on how to approach evaluation, the SPECS methodology (Jordanous, 2012a).

This chapter addresses the question: how should we evaluate the creativity of our computational creativity systems? It looks at how computational creativity as a research field has performed this task in the past, considering contextual and practical reasons that have historically affected evaluative practice.

In recent years, a more systematic approach to such evaluation has started to become more important. Various research has been proposed on how to evaluate the creativity of computational systems, but the computational creativity research community has not yet settled on a standardised approach. The chapter reflects on evaluative tools we have available to us and highlights the Standardised Procedure for Evaluating Creative Systems (SPECS) as a candidate set of guidelines for how to approach computational creativity evaluation. The question of how we choose an evaluation method, or how we judge the quality of our evaluation methods, is also addressed via five meta-evaluative criteria for judging what makes a good evaluation method (derived from the broader literature on good practice in evaluation).

Overall, this chapter argues that, as computational creativity researchers, we need to evaluate the creativity of our systems carefully if we are to justify labelling them as *creative* systems. Evaluation helps us understand how creative our systems actually are and how they could become more creative, and highlights what we have learned about creativity from our computational creativity research.

10.2 The Role of Evaluation and Why It Is Needed

> [U]nless the motivations and aims of the research are stated and appropriate methodologies and assessment procedures adopted, it is hard for other researchers to appreciate the practical or theoretical significance of the work. This, in turn, hinders . . . the comparison of different theories and practical applications . . . [and] has encouraged the stagnation of the fields of research involved.

Pearce, Meredith, and Wiggins (2002)

In 2002, Pearce et al. (2002) highlighted a 'methodological malaise' faced by those working with computational music composition systems due to a lack of methodological standards for development and evaluation of these systems, causing progress in this research area to 'stagnate'. It is not unimaginable that computational creativity research could succumb to this same malaise.

Evaluation is important for computational creativity research, allowing us to compare and contrast progress. Ignoring this evaluation stage deprives us of valuable analytical information about what our creative systems achieve, especially in comparison with other systems.

Currently many implementers of creative systems follow a creative-practitioner-type approach, producing a system and then presenting it to others, whose critical reaction determines its worth as a creative entity. A creative practitioner's primary aim, however, is to produce creative work to be judged on its own merits and quality, rather than to critically investigate creativity; in general, this latter investigative aim is important in computational creativity research. Creativity entails more than just the quality of the output; for example, what about novelty or variety? Yet computational creativity systems are often described as creative without justification for this claim. This cannot be the case for academic evaluation; the criteria by which you evaluate should be clearly stated, for rigour of approach and to enable comparison and criticism of evaluation results.

Evaluation helps identify where progress is being made and how the evaluated item can be improved upon. Without evaluation, how can research progress be demonstrated and tracked? And how can we understand and learn from our research without considering what has (and has not) been achieved?

Two different types of evaluation are summative evaluation and formative evaluation. In the context of this discussion, summative evaluation aims to provide a summary of a system's creativity, perhaps in quantitative form, for judgement of the amount of creativity demonstrated by that system. Formative evaluation, on the other hand, provides feedback on the system's strengths and weaknesses, to assist in improvements and developments during ongoing development or in reflections on how the system worked well and how it could be improved. The distinction between summative and formative evaluation is analogous to two types of educational feedback: giving a mark representing the work's quality, or giving constructive feedback on how to improve the work, respectively.[1]

Both types of evaluation have been examined in previous computational creativity evaluation literature. Colton, Pease, and Ritchie (2001) advocate an entirely formative approach aimed at using evaluation feedback 'in the design of creative programs rather than in the assessment of established programs' (Colton et al., 2001, p. 1), considering summative evaluation to be unnecessary. Ritchie (2007) focuses instead on summative evaluation to measure the creativity of an existing program, proposing

[1] As rightly pointed out by a reviewer of this text, summative feedback on a single system can also act as formative feedback for the field in general, as the field learns from the strengths and weaknesses of work done on a single system as a contribution to the broader research agenda overall.

criteria 'which give some indicators of the extent to which that program has been creative on that occasion.' (Ritchie, 2007, p. 74).

This work aims more towards the generation of formative feedback to help improve existing systems and design new systems. It does, however, recognise the value of some summative feedback, to reward particularly creative achievements with positive evaluations, identifying what progress systems are making and what contributions are being made to knowledge. Evaluation in this work is treated as reviewing systems, comparing them with others to an appropriate degree to see which are more creative than others and in what ways, and learning from this evaluation and comparison.

> It is important to be explicit . . . about the criteria that are being applied in making judgements of creativity. Ritchie (2001, p. 3)

Ritchie's point above stresses that it is not sufficient just to say that a system has been evaluated; how the evaluation was performed should be easily transparent, so the evaluation process can be repeated on other systems for consistency and for comparison of evaluation results. A transparent evaluation process also becomes available to others for critique and perhaps to learn from, helping us in sharing knowledge and learning from the work that we are doing. Sloman (1978) stresses that we need to be able to explain how our research results demonstrate some theoretical underpinning, not just assert our research findings without explanation of their wider significance. Such an explanation should not be contradicted by other existing, established theories or laws (except where existing theories are invalid and are disproved by the new theory). Lack of these explanations would, Sloman argues, affect the development of shared knowledge and understanding:

> Unfortunately, the role of such explanations in our thought is obscured by the fact that not everyone who requires, seeks or finds such an explanation, or who learns one from other people, asks this sort of question explicitly, or fully articulates the explanation when he has understood. Sloman (1978, p. 45)

To summarise: as a scientific research community, computational creativity researchers can benefit from more systematic/standardised approaches to evaluation of the creativity of our systems, to help us progress in understanding creativity and modelling it computationally.

10.3 Development of Creativity Evaluation Practices Over Time

Historically, the evaluation of the creativity of creative systems has, until recently, been relatively weak compared with what one might expect for a scientific discipline, though this situation has evolved recently as considerably more attention has been focused on this perspective of computational creativity research.

A 2011 survey (Jordanous, 2011) revealed issues with evaluative practice in computational creativity research at that time. To see how computational creativity systems were evaluated at that time, 75 journal and conference papers were surveyed,

with the aim of including all papers presenting a computational system that was described as being creative. Details of these sources are listed in Table 10.1.

Table 10.1 Computational creativity papers surveyed in the 2011 survey by Jordanous (2011). Papers were sourced from the journal special issues and conferences listed in the table. The papers listed as 'other relevant journal papers' were sourced using the *Web of Knowledge* and *Scopus* databases: a literature search was conducted to find all journal papers presenting details of a computational creativity system. Words and phrases such as 'computational creativity', 'creative system', 'creative computation', 'system' and 'creativity' were used as search terms.

Publication source	No. of papers presenting creative systems
2010 *Minds and Machines* **20**(4)	3
2009 *AI Magazine* **30**(3)	2
2006 *New Generation Computing* **24**(3)	4
2006 *Knowledge-Based Systems* **19**(7)	3
1996–2010 Other relevant journal papers	2
2010 ICCCX conference	25
2009 Dagstuhl seminar	10
2008 IJWCC08 workshop	12
2007 IJWCC07 conference	14
Total	75

Table 10.2 outlines the results of this survey. Out of 75 computational systems presented as being 'creative systems', 17 papers had no critical discussion or evaluation of the creative system. Of the 75 programs presented as creative systems, only a third (26 systems) were critically discussed in terms of how creative they were. Less than a quarter of the systems made any practical use of existing creativity evaluation methodologies.

Looking at the 18 papers that applied creativity evaluation methodologies to evaluate their system's creativity, no one methodology emerged as standard across the community. Colton's creative tripod framework (Colton, 2008) was used most often (six uses), with four papers using Ritchie's empirical criteria (Ritchie, 2007). No other methodology was used by more than one paper. Five papers proposed and used their own creativity metrics. Overall, in the 18 papers making practical use of evaluation methodologies, only 10 papers used those methodologies to evaluate how *creative* their systems were. The other papers adapted the chosen methodology to measure the *quality* of the systems (six papers), or proposed a methodology for creativity which actually only measures quality (two papers). Only a third of the systems (25 / 75) were evaluated by people other than the authors of the paper. There were only 11 examples of evaluative comparison between systems or against human performance (10 and one, respectively), to see if the system presented did outperform existing systems and if it represented any real research progress in the field. A further four papers included discussion of other related systems, but did not perform any evaluative comparisons.

Table 10.2 Summary of evaluation of the 75 creative systems surveyed

Paper makes at least a mention of evaluation	77%
Paper gives details of what evaluation has been done	55%
Paper contains section(s) on evaluation	51%
Paper states evaluation criteria	69%
Main aim of evaluation: creativity	35%
Main aim of evaluation: quality/accuracy/other	43%
Mention of creativity evaluation methodology	27%
Application of creativity evaluation methodology	24%
System compared with other systems	15%
System compared with systems by other researchers	11%
Systems evaluated by independent judges	33%

In general, journal papers tend to undergo a more stringent review process and be written and edited over a longer time frame than conference papers. One could hypothesise that if the focus of the survey was restricted only to journal papers, rather than also including conference papers, then a more systematic and scientific approach to evaluation might be found. To a certain extent, this hypothesis is validated; however, significant problems with the evaluations performed are still to be found. Taking the fourteen journal papers within the seventy five surveyed papers:

- Out of fourteen journal papers that present details of a computational creativity system, system evaluation is mentioned in almost all papers (twelve out of fourteen) and evaluation is actually performed and reported for all but one of these twelve.
- The transparency of the evaluation process is more pronounced for the set of journal papers, with elevent out of twelve journal papers making clear statements of the evaluation criteria being used for evaluation, and ten out of fourteen papers devoting a section of the paper to reporting methodological details and results of the system evaluation being done.
- Again, though, although each system is presented as being creative, not all systems are critically discussed in terms of their creativity. Only seven out of fourteen systems are evaluated on how creative they are, though at 50% this is at least an increase of 15% compared with the full set of seventy five papers included in the survey. The other 50% of systems are only evaluated in terms of accuracy and quality of performance.
- A standard creativity evaluation methodology is only mentioned in three of the fourteen papers. Of these three papers, only one paper uses a methodology for its intended purpose, of measuring creativity (using Ritchie's criteria (Ritchie, 2001)). The second of these three papers also mentions Ritchie's criteria but does not apply it practically for evaluation. The third paper uses Wiggins' proposals (Wiggins, 2006) to classify the computational system as creative but does not make any quantitative evaluation of how creative the system is.
- Only five systems are compared against the performance of other similar systems. Of these five, one paper compares its system with existing systems in the same

domain but not on the same criteria as are used for evaluation in that paper. Two more of these five papers compare the system reported with its research competition in terms of quality of performance but not in terms of creativity exhibited by the system.

10.3.1 Understanding the Survey Results

The survey findings reported above show that for creativity evaluation at that time, there was a lack of direction and standardisation of practice within the computational creativity research community – to the extent that creativity evaluation was often reported poorly or missed out entirely. But why was this the case? It is important to try and understand why this situation has developed within the computational creativity research community. Debates on such issues and problems had taken place often during these formative years of this research field, both in discussions between people in the field (A. Pease, personal communications, 2012) and in the context of relevant publications.

10.3.1.1 Definitional Difficulties in Evaluating Creativity

A perennial issue concerns the complex nature and difficulties of evaluating creativity. It is non-trivial to identify or measure creativity or many aspects related to creativity, for example aesthetic appeal, or the place of a product in the context of its domain. Creativity can involve contradictions, non-systematic processes and even flaws. (Traditionally, these are aspects which we may try to remove from our systems in the processes of development.) Creativity incorporates many different dimensions, and creativity may vary in how it is manifested in different domains, over different time periods and to different audiences (Jordanous, 2012b; Jordanous & Keller, 2016; Plucker & Beghetto, 2004).

A more practical approach can be afforded if one takes the view that creativity is multidimensional, with many factors contributing to the creativity of a creative system (Colton, 2008; Grace & Maher, 2019; Jennings, 2010; Jordanous, 2010b; Pease et al., 2001; Plucker, Beghetto, & Dow, 2004; Ritchie, 2007, 2019). This breaks down the concept of creativity to something more manageable and tangible, as opposed to an overarching, impenetrable concept of 'creativity'. Later in this chapter, a set of components of creativity are presented (see Fig. 10.1 or Jordanous and Keller (2016)) as a suggestion for how creativity can be characterised.

10.3.1.2 Conflicting Messages on the Importance of Evaluating the Creativity of Computational Creativity Systems

As the field has moved from its formative years of community development, into a position where it attracts enough research to support an annual international conference audience, journal special issues and soon (at the time of writing) a dedicated journal, evaluation has become more important for full research reports (A. Pease, personal communications, 2012). In peer review instructions for research events in the field (see below), lack of evaluation in a paper has become a valid reason for rejecting a long paper or changing its status to that of a position paper (representing that work on the system is not yet complete).

Until relatively recently, however (Jordanous, 2011), it was possible to publish papers on computational creativity without including any real evaluation. Below a survey of evaluation in computational creativity papers up to 2010 (Jordanous, 2011) is described, which revealed a lack of emphasis on scientific or systematic evaluation practices up to that time. Instead, many researchers were taking an approach that to a certain extent follows the common practice of creative practitioners: produce work and then exhibit it to an audience, whose reaction (both immediate and longer-term) asserts the value of the work, instead of performing retrospective comparative analysis of the creativity of the work.

This practitioner-based approach was not endorsed in calls for papers for computational creativity research events. Creativity evaluation metrics and strategies have frequently appeared on the list of topics of interest for workshops and symposiums in the form of phrases such as 'Evaluation of Creativity' in the Workshops on creative systems, 2002–04; 'the assessment of creativity in AI programs' in the AISB Workshop, 2003; 'how we assess creativity in computers' in IJWCC 2007, 'Metrics, frameworks and formalizations for the evaluation of novelty and originality'[2] in the Computational Creativity Workshop 2005; and the rephrasing 'Metrics, frameworks and formalizations for the evaluation of creativity in computational systems' in the Computational Creativity Workshops, 2006 and 2008. This last wording has appeared in the call for papers for all ICCC conferences to date (2010–2016). (For ICCC'11 only, this phrasing appeared with a qualifier: 'quasi-formal approaches that, for example, argue for recognition without definition or that define the absence of creativity may have interesting implications for computational creativity', possibly in response to work on evaluation by Colton (2008).

With these conflicting messages on evaluation, another reason for the lack of consistency in creativity evaluation practice may have been that reviewers' judgements and decisions may be influenced by personal criteria that may differ from reviewer to reviewer (R. Pérez y Pérez, personal communications, 2012), hence the emphasis on a need for evaluation may also differ across reviewers. The variety of disciplinary backgrounds that reviewers may come to computational creativity from makes Pérez y Pérez's point particularly pertinent. Additionally, this culture may have developed

[2] The combination of novelty and originality is often used as a reductionist definition of creativity (Brown, Boden, D'Inverno, & McCormack, 2009; Pease et al., 2001; Pereira & Cardoso, 2006; Ritchie, 2007).

as a side effect of the inclusive efforts to build up a community of computational creativity researchers; decisions on whether a paper should be accepted for a conference could be based around whether the paper would trigger interesting debate, rather than how academically rigorous its presentation was (A. Pease, personal communications, 2012). This was enhanced by the practice, since 2001, of accepting position/short papers (reports of work in progress or comments on research directions) alongside technical/long papers (detailed technical reports of creative systems or foundational theory). Whilst more thorough academic reporting is required for technical papers, a requirement which has been particularly imposed in the last few years (A. Pease, personal communications, 2012), position papers often report work in progress rather than completed work, and so have fewer requirements imposed for academic rigour in reporting. Position papers encourage the reporting of current work and new methods even if the work is not yet fully completed. Unfortunately, those proceedings often do not clearly distinguish technical papers from position papers (for example, the proceedings of ICCC'10 and ICCC'11, where a position paper is distinguishable from a full technical paper only by its number of pages). Hence this differentiation in quality can easily be missed, making a lack of evaluative (and other academic) rigour seemingly more acceptable in this community.

Although attitudes towards creativity evaluation have largely changed, a few practical or conceptual issues complicating evaluation do exist within the field, which will now be considered.

10.3.1.3 Difficulties in Finding Relevant Systems for Comparison

The above survey criticised situations where systems are evaluated without contextual references to the context of research progress in that area as a whole. If a researcher is unaware of related work by others, then they cannot learn from that work and their own work is potentially less likely to make a highly relevant contribution to the advance of research in that particular field. Activity in the field of computational creativity research has increased greatly over the last decade or so, as shown by the progress in research events and the recent journal special editions focused on this area (see Table 10.1). Many creative systems have now been developed (for example, 75 reports of systems were analysed in the aforementioned survey). If similar systems exist and a body of research builds up in a particular area, then it is useful to consider how research in that area is progressing collectively. This benefit has been demonstrated in research into narrative/story-generation systems (Gervás, 2009), a long-standing and thriving research area within computational creativity research (e.g. Bringsjord (2000), Meehan (1981), Peinado, Francisco, Hervás, and Gervás (2010), Peinado and Gervás (2006), Pérez y Pérez and Sharples (2004), Tearse, Mawhorter, Mateas, and Wardrip-Fonin (2011), Turner (1994)).

Is the central aim of a creative system to generate products that no other systems generate, or to generate behaviour distinct from all other systems? If a system is unique, then how can it be compared with other systems? But how would one be able

to find out if a system is producing unique results, if its output is not compared with other systems?

There are situations where a creative system operates in a niche where no other system exists. For example, ERI-Designer is believed to be the sole exemplar system of creativity in furniture arrangement (Aguilar, Hernandez, Pérez y Pérez, Rojas, & Zambrano, 2008; Pérez y Pérez, Aguilar, & Negrete, 2010). In these cases, direct comparisons between two equivalent systems cannot be made; however, comparisons could be made between the system and humans performing the same task (as in Pérez y Pérez et al. (2010)), or with a considered comparison of the appropriate crossovers with systems operating in a reasonably similar domain. Take the example of ERI-Designer, mentioned above: it could perhaps be compared to architectural design systems or game design systems.

A broad evaluation from a wide perspective can be performed on systems which are fundamentally different; we can learn both from the evaluation results and by understanding the ways in which the systems are different. Distinctions have been drawn between evaluation of one system as a single research project and the evaluation of progress in a particular strand of research.

There are some types of systems that are so fundamentally different that there is no area of crossover to compare; however as later discussions in this chapter will point out, some aspects of creativity are universal across different systems. While the amount of crossover between system domains determines the extent to which the systems can be compared, we do not need to be restricted to evaluating systems that are very similar before meaningful comparisons start to emerge.

Different versions of the same system can also be compared, to see what improvements have been made and to measure progress. Some of the papers reviewed in the above survey did extend and develop existing systems, for example R. P. Whorley, Wiggins, and Pearce (2007), R. Whorley, Wiggins, Rhodes, and Pearce (2010), papers related to MEXICA and its variants (Gervás & Pérez y Pérez, 2007; Montfort & Pérez y Pérez, 2008; Pérez y Pérez et al., 2010), and papers related to ERI-Designer (Aguilar et al., 2008; Pérez y Pérez et al., 2010). Generally, comparisons between different versions of the system were limited, but were present to some degree.

10.3.1.4 Different Types of Evaluation

In computational creativity evaluation, a wide variety of different types of methods have been considered. Eigenfeldt, Burnett, and Pasquier (2012) present different ways in which a computational creative system can be evaluated: evaluative experiments, peer evaluation, critical evaluation, audience reaction and academic paper acceptance. The survey by Jordanous (2011) took an inclusive view of evaluation and included scenarios where the system was presented to an audience and audience feedback was obtained. A similar variety of points of views was acknowledged during discussions on evaluation at the 2009 Dagstuhl seminar on computational creativity, including the perspectives of ‘viewer/experiencer’, ‘creator’ and ‘interactive participant’ (Brown et al., 2009, p. 1). This present chapter concentrates on post-implementation evaluation

of a system, but incorporates within that scope the possibility of peer evaluation, expert evaluation and audience evaluation as options for such evaluation.

10.3.1.5 Formative Versus Summative Evaluation

Can you give a computational creativity system a summative evaluation, such as a 'creativity score'? Although this might seem appealing from a scientific metrics point of view, in practice it is not so useful to format evaluation feedback in such a summative way. Formative feedback is a useful result of evaluation: discussions on evaluation at the 2009 Dagstuhl seminar on computational creativity argued that '[e]valuation can feed back into the system to affect (hopefully improve) future performance' (Brown et al., 2009, p.1). It seems somewhat contradictory to identify what constitutes creativity, but then sacrifice more detailed qualitative feedback on each component for a single score.

10.3.1.6 Why Not Just Ask Humans How Creative Our Systems Are?

Experiments can be run with human judges to evaluate the creativity of a system, as was done in the evaluation case study described later in this chapter. There is definitely a place for soliciting human opinion in creativity evaluation, not least as a simple way to consider the system's creativity in terms of those creative aspects which are overly complex to define empirically, or which are most sensitive to time and current societal context. The process of running adequate evaluation experiments with human participants, though, takes up a good deal of time and effort. Human opinion is variable; what one person finds creative, another may not (Jennings, 2010; León & Gervás, 2010). Therefore large numbers of participants may be required, to capture a general consensus of opinion.

In addition to the time and resources necessary to devise and run suitable evaluation experiments with large numbers of people, extra issues such as the procedure of applying for ethics permissions are introduced. There may also be some difficulty in attracting suitable participants, and a cost associated with paying participants. These issues may have adverse effects on the research process, many of which are out of our direct control to resolve. It would be useful if this outlay of research time and effort could be reduced.

There are other practical concerns which hinder us from using human judges as the sole source of evaluation of a system. Human evaluators can say whether they think something is creative but can usually give minimal insight into why it is creative. As mentioned above, it is hard to define why something is creative; this is a tacit judgement rather than one we can easily voice. It is useful to have a more informed idea of what makes a system creative, to understand both why a system is creative and what needs to be worked on to make the system more creative.

Here one must acknowledge a common problem in computational creativity research: human reluctance to accept the concept of computers being creative. On the

other hand, researchers keen to embrace computational creativity may be positively influenced towards assigning a computational system more credit for creativity than it perhaps deserves. Hence our ability to evaluate creative systems objectively can be significantly affected once we know (or suspect) we are evaluating a computer rather than a human.

10.3.1.7 Issues in Providing a Standard Tool Across Creativity

Jordanous (2012b) argues that what is needed at this stage of the field's development is an established and standardised approach to evaluation for tracking and evaluating progress in computational creativity. How can an evaluative approach for researchers be flexible enough to be adopted for several different types of creativity, without falling into a trap of being all-inclusive but being virtually useless for definitional purpose and any real information?

By its very nature, creativity manifests itself in a variety of forms, with different creative domains prioritising aspects of creativity differently (Jordanous, 2012b). For the same reason, though, some standardisation is necessary to avoid the concept of creativity being interpreted too liberally, where any system could be argued to be creative depending on how creativity is characterised. This approach requires that the standards used to judge creativity are stated and open to discussion.

Various researchers have offered a standardisable methodology that can be parameterisable and customisable, allowing an appropriate level of domain-specific detail to be given, without being overly restrictive or abstract. This affords us greater consistency across research as we can use benchmarks and comparisons to help gauge our progress.

Pease, Colton and colleagues have proposed the FACE (Frame, Aesthetic, Concept, Example) and IDEA (Iterative–Development–Execution–Appreciation) models for characterising and describing creative systems (Colton, Goodwin, & Veale, 2012; Pease & Colton, 2011). Ritchie proposes empirical criteria to assess the creativity of a system based on rating the system's products for how typical of the intended genre they are and for the value of the products (Ritchie, 2007). Colton offers a *creative tripod* framework to evaluate whether a system is a candidate for being considered creative (Colton, 2008): does the system demonstrate skill, imagination and appreciation? Pease et al. (2001) describe various tests of a creative system's output, input and creative process.[3]

Although not without flaws, the methods mentioned above offer useful insight for evaluative purposes, for example in the way the concept of creativity is broken down into constituent components and the suggestion of practical tests to carry out in evaluation. Despite these methods being available, though, no method has yet been adopted as standard evaluative practice by the research community. The next section looks at the Standardised Procedure for Evaluating Creative Systems (SPECS)

[3] The chapter by Graeme Ritchie in this volume (Ritchie, 2019) explores the details of different evaluation methods currently available.

approach, a methodology guiding us more broadly in how to approach creativity evaluation using the knowledge and tools we have within creativity research.

10.4 Standardising Our Approach to Evaluation

The *Standardised Procedure for Evaluating Creative Systems* (SPECS) forms an overarching set of guidelines for how to tackle evaluation of creative systems.

SPECS is a set of methodological steps to perform, to evaluate the creativity of creative systems. Each step is introduced and discussed individually below, with reference to two case studies in Jordanous (2012b) that exemplify the use of SPECS for evaluation. The methodology is presented in full in Table 10.3. Elsewhere, Jordanous (2012b) gives a practical 'walkthrough' of the steps of SPECS, in the form of interlinked decision-tree-style diagrams that illustrate different paths through the application of SPECS.

Table 10.3 SPECS, the Standardised Procedure for Evaluating Creative Systems

1. *Identify a definition of creativity that your system should satisfy to be considered creative.*
   a. What does it mean to be creative in a general context, independent of any domain specifics?
      - Research and identify a definition of creativity that you feel offers the most suitable definition of creativity.
   b. What aspects of creativity are particularly important in the domain your system works in (and what aspects of creativity are less important in that domain)?
      - Adapt the general definition of creativity obtained from Step 1a so that it accurately reflects how creativity is manifested in the domain your system works in.
2. *Using Step 1, clearly state what standards you use to evaluate the creativity of your system.*
   - Identify the criteria for creativity included in the definition obtained from Step 1 (a and b) and extract them from the definition, expressing each criterion as a separate standard to be tested.
3. *Test your creative system against the standards stated in Step 2 and report the results.*
   - For each standard stated in Step 2, devise test(s) to evaluate the system's performance against that standard.
   - The choice of tests to be used is left up to the individual researcher or research team.
   - Consider the test results in terms of how important the associated aspect of creativity is in that domain, with more important aspects of creativity being given greater consideration than less important aspects. It is not necessary, however, to combine all the test results into one aggregate score of creativity.

10.4.1 Step 1: Defining Creativity

Identify a definition of creativity that your system should satisfy to be considered creative:

(a) What does it mean to be creative in a general context, independent of any domain specifics?
(b) What aspects of creativity are particularly important in the domain your system works in (and what aspects of creativity are less important in that domain)?

10.4.1.1 Identifying General Aspects of Creativity

Step 1a of SPECS asks the researcher to identify a general definition of creativity. As Jordanous (2012b) notes, there are a plethora of definitions of creativity and choosing an appropriate definition is non-trivial. Jordanous and Keller (2016) offer a general and interdisciplinary characterisation of creativity, empirically derived using cognitive linguistics from a corpus of texts about creativity. This characterisation is in the format of a set of 14 components that act as building blocks for creativity, pictured in Fig. 10.1 and defined further elsewhere (Jordanous, 2012a, 2012b; Jordanous & Keller, 2016). These components represent common themes across creativity in general and can be used to fulfil the requirements of Step 1a.

10.4.1.2 Identifying the Relative Contributions of Different Aspects in a Creative Domain

While some aspects of creativity are shared by all types of creativity, some aspects of creativity will be prioritised (or deprioritised) more in some domains than others (Jordanous, 2012b; Plucker & Beghetto, 2004). For Step 1b, the researcher needs to be aware of how creativity is demonstrated in the creative domain they are focusing on, adjusting the definition obtained from Step 1a accordingly.

It is recommended that for Step 1a, the components of creativity pictured in Fig. 10.1 (Jordanous & Keller, 2016) are used. If this is the case, the researcher should investigate the relative important of each component in their particular domain, and weight the contribution of each component accordingly. This may be done by quantifying the importance of each component. For example in Case Study 1 of Jordanous's evaluation (Jordanous, 2012b), components were weighted by how often they were mentioned in written discussions about musical improvisation (Jordanous & Keller, 2012). Alternatively, the components could be categorised according to level of importance for that domain, as was demonstrated in Jordanous's Case Study 2 (Jordanous, 2012b), where components were classified either as *Crucial for creativity*, *Quite important*, *A little important* or *Not at all important*.

The importance of each component can be investigated in many ways, such as consulting the opinion of experts and/or the general public, analysing prior research

Fig. 10.1 The 14 components of creativity: a general and interdisciplinary characterisation of creativity, empirically derived from discussions about the nature of creativity, using corpus linguistics and cognitive linguistics (Jordanous & Keller, 2016). Collectively, these components represent common themes across creativity in general and can be used to fulfil the requirements of Step 1a of SPECS.

or consulting general knowledge about that field. The researcher is advised to support their choices using relevant knowledge.

As the components of creativity (Jordanous & Keller, 2016, Fig. 10.1) were derived from general discussions about creativity, and identify common general themes across creativity, it is strongly recommended that these components be used for Step 1a of SPECS (what does it mean to be creative in general?). The components can then be customised according to importance during Step 1b (what is more/less contributory to creativity in a specific creative domain of interest?). If one chooses a different interpretation of creativity, this choice should be clearly stated and justified as to why it forms a base definition of creativity, both in general and for the particular domain of interest.

10.4.2 Step 2: Identifying Standards to Test the System's Creativity

Using Step 1, clearly state what standards you use to evaluate the creativity of your system.

For Step 2 of SPECS, the definition of creativity obtained from Step 1 is transformed into an equivalent (or as close as possible) set of operational standards, for practical application to testing of the system. If one is using the 14 components of creativity in Fig. 10.1 (Jordanous & Keller, 2016), each component becomes an aspect of the system to be tested. Little analysis is required here, except perhaps to re-express the components in a form more relevant to that particular domain, as is done in Jordanous's Case Study 1 (Jordanous, 2012b).

Further analysis may be required, if one is using other definitions, to convert the definition into standards for objective testing. In prose definitions, the conversion from definition to standards is not so direct. Take, for instance, 'Creativity is the ability to come up with ideas or artefacts that are *new*, *surprising and valuable*' (Boden, 2004, p. 1): does Boden require a system to actually produce these ideas/artefacts before it can be deemed creative, or merely have the ability to do so? Careful analysis of the specific definition is needed.

No further detail is given in this present discussion on heuristics for this conversion; this process will be specific to the definition used. Instead this acts as another reason for recommending the components shown in Fig. 10.1 for Step 1.

10.4.3 Step 3: Testing Systems Using the Components

Test your creative system against the standards stated in Step 2 and report the results.

The choice of what tests to use for evaluation is heavily dependent on the standards chosen to be tested and the preferences, capabilities and equipment/facilities of the researcher(s) involved. Several issues surround this choice of tests (discussed in detail by Jordanous (2012b)), which should be taken into account at this stage:

- the product/process debate in computational creativity evaluation;
- Boden's distinction between P-creativity and H-creativity;
- practical issues in using human judges;
- the expertise and bias of the evaluators;
- using quantitative and qualitative methods.

As the purpose of SPECS is to provide detailed feedback on system performance rather than an overall 'creativity score', it is not necessary to recombine the test results for a single aggregated measure of the whole system. One should, however, consider individual test results relative to the importance of individual aspects, with results for more important aspects for a particular domain being given more emphasis than less important aspects. As an example, two different ways of identifying and weighting component importance were explored in the two case studies presented by Jordanous (2012b): one with numerical weighting of importance, and another with categorical weighting.

10.4.4 The Intention of the SPECS Approach

The aim of this approach is to encourage a more systematic overarching approach to computational creativity evaluation, one that incorporates the flexibility needed for computational creativity research. This approach aims to examine the creativity of a creative system more systematically; to pinpoint why and in what ways a system can justifiably be said to be creative. The point is to understand to a greater level of detail exactly why a system can be described as creative. The SPECS approach enables us to investigate in what ways a system is being creative and how research is progressing in this area, using an informed, multifaceted approach that suits the nature of creativity.

SPECS allows comparison between a creative system and other similar systems. A clear statement of evaluation criteria makes the evaluation process more transparent and makes the evaluation criteria available to other researchers, avoiding unnecessary duplication of effort. By using the same evaluation standards across different systems, we can see where systems' strengths and weaknesses are.

It should be acknowledged that a creative system is evaluated according to standards at a particular point in time, where a creative domain is in a certain state, viewed by society in a certain context. These standards may change over time. If similar systems have previously been presented to similar audiences at similar times, however, then the evaluation standards can be reused. Hence detailed comparisons can be made using each standard, to identify areas of progress. For example, in a comparison of the GAmprovising musical improvisation system (Jordanous, 2010a) against two other musical improvisation systems, it was found that improving the interactive/communicative aspects, the musical information and the ability to demonstrate some sort of intent or drive to improvise would have a large impact in improving

the creativity of the GAmprovising system compared with the other two systems (Jordanous, 2012b).

10.4.5 What SPECS Is Not

SPECS does not offer a 'measurement system' that finds the most creative system, or gives a single summative rating for a creativity system (though people may choose to use and adopt the approach for these purposes if it is relevant in their domain). Such a scenario is usually impractical for creativity, both human and computational. There is little value in giving a definitive rating of computational creativity, especially as we would be unlikely to encounter such a rating for human creativity.

Nor is this an attempt to dissuade researchers from attempting to implement creative systems, or to put obstacles in the way of such researchers such that they are forced to target other goals and justifications for their research rather than the pursuit of making computers creative. It is of course reasonable for computational creativity researchers to aim their work towards better understanding of creativity, rather than to implement computational systems that are themselves creative. For example, the pursuit of making the YQX music performance system creative (Widmer, Flossmann, & Grachten, 2009) was 'abandoned' in favour of exploring human creativity via those authors' research. However for those researchers whose intention is to implement a computer system which is creative, the approach outlined in this chapter offers a methodological tool to assist progress.

10.4.6 Incorporating Other Evaluation Frameworks

Within SPECS, different evaluation approaches and frameworks can be applied. SPECS lets the evaluator choose the most appropriate existing evaluation suggestions, without being tied into a fixed definition of creativity that may not apply fully in the domain they work in.

Depending on how creativity is defined by the researcher(s), previous evaluation frameworks (Colton, 2008; Grace & Maher, 2019; Pease et al., 2001; Ritchie, 2007, 2019, and other discussions) may be accommodated if appropriate for the standards by which the system is being evaluated. For example, if skill, appreciation and imagination are identified as some key components of creativity for a creative system, it would be appropriate to use the creative tripod (Colton, 2008).

SPECS does not impose specifications of what tests to include. What is emphasised here is that, for scientific evaluation, we must clearly justify claims for the success or otherwise of research achievements. This approach affords such clarity.

10.4.7 Key Standardised Aspects of SPECS

It is emphasised that at each stage of the evaluative process, evaluators are expected to state what is being done or chosen and to justify why such actions are being taken or decisions made (with supporting evidence where appropriate), for a clear and reasoned evaluative process. Such transparency contributes to an overarching aim of SPECS: to encourage standardised, relevant and repeatable evaluation processes to be proposed, critiqued and developed by communications and progress in the field as a whole.

Additionally in this vein, before conducting SPECS, evaluators should be aware of any previous approaches to evaluating the creativity of comparable systems. Should any such creativity evaluations exist, or if the evaluators wish to apply an existing evaluation method, the evaluation method should be critically reviewed as to whether it is appropriate for use to evaluate the current system(s) (or to complement a larger evaluation). This critical review should consider the following points (and may require evaluators/peer critics to conduct Step 1 (both Steps 1a and 1b) to inform their choices):

- Does the previous approach actually evaluate creativity?
- Is that approach suitable for the type of creativity you are evaluating (either as it is or with some customisation)?
  - Can the previous representation of creativity be applied directly to your system's domain?
    - Does that creativity representation accommodate general aspects of creativity? (Carry out Step 1a if necessary to inform this decision.)
    - Does that representation take into account what we know about that type of creativity? (Carry out Step 1b if necessary to inform this decision.)
  - If that previous representation of creativity is not appropriate as is for your system, can it be customised to your domain successfully, without missing important details or being too general or too specific?
    - Is the definition overly specific to the system that it was applied to, or to its domain, without taking into account more general aspects of creativity? (Carry out Step 1a if necessary to inform this decision.)
    - Is the definition overly general, not taking into account specific requirements for creativity in that domain? (Carry out Step 1b if necessary to inform this decision.)
- Was the previous approach suitable for the type of creativity the original evaluators were evaluating?[4]
- Similarly, can the previous evaluation methods be applied directly or customised in an appropriate way for your system?

[4] This question focuses more on the appropriateness of the previous evaluation and only indirectly assists the current evaluation task; however, after investigating the existing evaluation, the current evaluator(s) will be well placed to provide formative feedback to the original evaluators, as a valuable peer-review contribution.

The considered critique and reuse of suitable existing evaluation approaches and models emphasise the overriding aim towards a standardised approach to evaluation of computational creativity that encourages comparison and the placing of an individual system(s) within a wider research context, as measures of progress in the research area and the field as a whole.

10.5 Evaluating Creativity Evaluation Methods

As we see above, there are a number of ways in which computational creativity researchers can evaluate the creativity of their software, either within the SPECS approach or in stand-alone evaluation. But which should computational creativity researchers use?

One should note here that we are unlikely to find one single fully specified, detailed, step-by-step methodology to suit all types of creative system. What we can do is to understand the strengths and weaknesses of different methodologies. Through application and comparison between different methodologies, we can refine and develop our evaluation strategies within computational creativity so that we can mutually learn from our advances and mistakes: the very essence of what evaluation offers researchers, after all.

How can these methodological tools for evaluation be compared against each other? Reviewing various features of the methodologies and comparing them against each other helps us to learn through comparison. Five meta-evaluation standards have been identified for comparison and evaluation of creativity evaluation methodologies, drawn from cross-disciplinary reviews of evaluative practice (Jordanous, 2014):

- *Correctness*: how accurately and exhaustively the evaluation findings capture and represent the actual system and its performance.
- *Usefulness*: how informative the evaluative findings are for understanding and potentially improving the creativity of the system.
- *Faithfulness as a model of creativity*: how faithfully the evaluation methodology evaluates the *creativity* of a system (as opposed to other aspects of the system).
- *Usability of the methodology*: the ease with which the evaluation methodology can be applied in practice, for evaluating the creativity of systems.
- *Generality*: how generally applicable this methodology is across various types of creative systems.

With these meta-evaluation criteria, we can now compare evaluation results obtained through different methods and discuss how useful each of these evaluations are to the computational creativity researcher. Gathering effective evaluative feedback, using solidly developed evaluation methodologies, assists further development of computational creativity research and helps identify more clearly the contributions to knowledge made by our research.

The meta-evaluation standards above were applied in a practical case study (Jordanous, 2012b, 2014) to evaluate various different evaluation methodologies. Three

different musical improvisation computer systems were evaluated to consider how creative each system was. Five methods were compared in the case study: four computational creativity evaluation methodologies and a survey of human opinion on how creative a system was judged to be by several human judges. The case study reported in Jordanous (2014) considered how well the creativity evaluation methodologies performed for this assessment. This case study helped us appreciate the strengths and weaknesses of each creativity evaluation methodology as used in this context, guiding us in our evaluative choices when developing computational creativity research. The evaluation methodologies included in this study were[5] (listed in chronological order):

- Ritchie's set of formal empirical criteria for use in assessing creativity of a system based on its products (Ritchie, 2001, 2007).
- Colton's Creative Tripod framework of skill, imagination and appreciation (Colton, 2008).
- The FACE (Frame, Aesthetic, Concept, Example) model (Colton et al., 2012; Pease & Colton, 2011).
- SPECS methodology described above and in Jordanous (2012a, 2012b), applied using the components of creativity presented by Jordanous and Keller (2016) that are shown in Fig. 10.1. (This application was referred to as 'SPECS+cc' in the case study.)

Evaluators (developers of musical improvisation systems) were asked to view all the evaluative feedback obtained in Case Study 1 of Jordanous (2012b). They were then asked to give their opinions on various aspects of each methodology and on the results obtained. Finally, the evaluators were asked to rank the evaluation methodologies according to how well they thought the methodologies evaluated the creativity of their system overall.

Overall, the application of SPECS+cc and Ritchie's empirical criteria compared favourably with the other methodologies. SPECS+cc performed well on most of the five meta-evaluation criteria, though the volume of data produced by SPECS+cc raised questions about SPECS+cc's usability compared with more succinct presentations. Colton's creative tripod was the easiest to use, although there were some concerns about the generality of the tripod across creative domains and its faithfulness as a general model of creativity. Ritchie's criteria were considered accurate but there were usability issues with the abstract nature of the criteria and accompanying function definitions. The FACE model was considered quite user-friendly but perhaps limited in how it could incorporate aspects of creativity that were important to the system domain but outside of the FACE model. Each of the evaluation methodologies proved to be an improvement (in at least some ways) over the approach of simply asking people's opinions on how creative the systems were.

How useful is this feedback? The underlying aim of this meta-evaluative case study was to focus on how well the SPECS methodology (Jordanous, 2012b) performed when applied using the components of creativity shown in Fig. 10.1 (Jordanous & Keller, 2016). These results were too small in scope to be a comprehensive

[5] The chapter by Graeme Ritchie in this volume (Ritchie, 2019) details each of these methodologies.

evaluation of the evaluation methodologies covered, but they do help to give us some initial guidance. Feedback was gained as to how to improve SPECS+cc and what its strengths were in comparison with other methods.

Considering all the observations made in this chapter from the perspective of the five meta-evaluation criteria presented here, SPECS+cc performed well in comparison with the other evaluation methodologies with respect to its faithfulness in modelling creativity. SPECS+cc also performed better than Ritchie's criteria for usefulness and correctness and produced larger quantities of useful feedback than Colton's creative tripod (because less information was collected for Colton's creative tripod). A consequence of the information collection meant that Colton's creative tripod was the easiest to use of the methodologies evaluated.

Somewhat counterintuitively, meta-evaluation revealed that all the methodologies were more likely to generate reliable results about creativity, compared with the difficulties encountered if one ignored these methodologies and instead surveyed human opinion on how creative the systems were. A number of participants in the opinion surveys reported that they evaluated systems based on factors other than creativity, owing to difficulties in evaluating creativity of computational systems without a definition of creativity to refer to. There is also some question about whether human opinion surveys could be carried out to evaluate all types of creativity (particularly where creativity is not manifested outwardly, in the production of artefacts); this affects the general applicability of using opinion surveys. Reliance on the existence of output examples also affects the usability and generalisability of Ritchie's criteria.

10.6 Concluding Remarks

A systematic approach to the evaluation of creativity is essential for progress in computational creativity. Surveying the literature on computational creativity systems in 2011, evidence suggested that scientific evaluation of creativity had been neglected (for various reasons), although this trend is now reversing as more emphasis is placed on the importance of evaluation.

It is important not to ignore a key issue in research merely because it is tricky to tackle. Instead, we should acknowledge the issue and surrounding debate and look for a 'working' answer if necessary, simplifying our assumptions to make the issue more tractable. Taking a practical and proactive approach reduces perceived barriers to research, making 'hard' issues such as evaluation of creativity more manageable and less stifling. Performing evaluation, based on a working understanding of creativity if necessary, provides more informative, more useful and more deeply grounded contributions than performing no evaluation at all.

The development of creativity evaluation methods has become a key current area of interest in the computational creativity research community, as partly illustrated by the prominent inclusion of requests for papers on evaluation in each year's calls for papers for ICCC conferences. Several methods have been proposed for

evaluating creativity. In particular, the Standardised Procedure for Evaluating Creative Systems (SPECS) is an overarching set of guidelines for how to approach evaluating computational creativity: identify (and clearly state) what it means for your system to be creative, identify evaluative standards which represent this characterisation of creativity, and devise/implement tests to evaluate your system accordingly.

For the purposes of progressing in research, learning from advances and improving what has been done, how useful are these different evaluation methodologies? To assist in answering this question, five meta-evaluation criteria are listed below, taken from a cross-disciplinary review of good practice in evaluation of areas relevant to computational creativity research (Jordanous, 2014). These five criteria help us to evaluate the evaluation methodologies themselves (for example, as described in the case study cited above) to elicit feedback on the relative strengths and weaknesses of each approach:

- correctness;
- usefulness;
- faithfulness as a model of creativity;
- usability of the methodology;
- generality.

Taken together, the considerations addressed in this chapter have helped us explore best practice in computational creativity evaluation. These explorations help us develop the tools we have available to us as computational creativity researchers. What David Chalmers says when introducing his work on the study of consciousness theory can be applied equally well to this study of creativity evaluation:

> if we are to make progress, the first thing we must do is face up to the things that make the problem so difficult. Then we can move forward toward a theory, without blinkers and with a good idea of the task at hand. Chalmers (1996, p. xii)

Acknowledgements Thanks to Alison Pease, Steve Torrance and Nick Collins, Al Biles, Bob Keller, George E. Lewis, Chris Thornton, Chris Kiefer, Gareth White and Jens Streck for helpful comments during the formulation of these ideas.

References

Aguilar, A., Hernandez, D., Pérez y Pérez, R., Rojas, M., & Zambrano, M. d. L. (2008). A computer model for novel arrangements of furniture. In *Proceedings of the 5th International Joint Workshop on Computational Creativity* (pp. 157–162). Madrid.

Boden, M. A. (2004). *The creative mind: Myths and mechanisms* (2nd). London: Routledge.

Bringsjord, S. (2000). *Artificial intelligence and literary creativity: Inside the mind of BRUTUS*. London: Lawrence Erlbaum Associates.

Brown, D., Boden, M., D'Inverno, M., & McCormack, J. (2009). Computational Artistic Creativity and its Evaluation. In M. Boden, M. D'Inverno, & J. McCormack (Eds.), *Computational creativity: An interdisciplinary approach* (09291, pp. 1–8). Dagstuhl Seminar Proceedings. Dagstuhl, Germany.

Chalmers, D. J. (1996). *The Conscious Mind*. New York: Oxford University Press.

Colton, S. (2008). Creativity versus the Perception of Creativity in Computational Systems. In *Proceedings of AAAI Symposium on Creative Systems* (pp. 14–20). Stanford, CA.

Colton, S., Charnley, J., & Pease, A. (2011). Computational Creativity Theory: The FACE and IDEA descriptive models. In *Proceedings of the 2nd International Conference on Computational Creativity* (pp. 90–95). Mexico City.

Colton, S., Goodwin, J., & Veale, T. (2012). Full-FACE poetry generation. In *Proceedings of the 3rd International Conference on Computational Creativity* (pp. 95–102). Dublin.

Colton, S., Pease, A., Corneli, J., Cook, M., & Llano, M. T. (2014). Assessing progress in building autonomously creative systems. In *Proceedings of the 5th International Conference on Computational Creativity*, Ljubljana, Slovenia.

Colton, S., Pease, A., & Ritchie, G. (2001). The Effect of Input Knowledge on Creativity. In *Proceedings of workshop program of ICCBR-Creative Systems: Approaches to Creativity in AI and Cognitive Science*. Retrieved from http://citeseerx.ist.psu.edu/viewdoc/download?doi=10.1.1.124.1377%5C&rep=rep1%5C&type=pdf

Eigenfeldt, A., Burnett, A., & Pasquier, P. (2012). Evaluating musical metacreation in a live performance context. In *Proceedings of the 3rd International Conference on Computational Creativity* (p. 140). Dublin.

Gervás, P. (2009). Computational Approaches to Storytelling and Creativity. *AI Magazine*, *30*(3), 49–62.

Gervás, P., & Pérez y Pérez, R. (2007). On the fly collaborative story-telling: Revising contributions to match a shared partial story line. In *Proceedings of the 4th International Joint Workshop on Computational Creativity* (pp. 13–20). London.

Grace, K., & Maher, M. L. (2019). Expectation-based models of novelty for evaluating computational creativity. In T. Veale & F. A. Cardoso (Eds.), *Computational creativity: The philosophy and engineering of autonomously creative systems* (pp. 193–207). Springer.

Jennings, K. E. (2010). Developing Creativity: Artificial Barriers in Artificial Intelligence. *Minds and Machines*, *20*(4), 489–501.

Jordanous, A. (2010a). A Fitness Function for Creativity in Jazz Improvisation and Beyond. In *Proceedings of the International Conference on Computational Creativity* (pp. 223–227). Lisbon.

Jordanous, A. (2010b). Defining Creativity: Finding Keywords for Creativity Using Corpus Linguistics Techniques. In *Proceedings of the International Conference on Computational Creativity* (pp. 278–287). Lisbon.

Jordanous, A. (2011). Evaluating Evaluation: Assessing Progress in Computational Creativity Research. In *Proceedings of the 2nd International Conference on Computational Creativity, ICCC-11*, Mexico City.

Jordanous, A. (2012a). A Standardised Procedure for Evaluating Creative Systems: Computational Creativity Evaluation Based on What it is to be Creative. *Cognitive Computation*, *4*(3), 246–279.

Jordanous, A. (2012b). *Evaluating Computational Creativity: A Standardised Procedure for Evaluating Creative Systems and its Application* (Doctoral dissertation, University of Sussex, Brighton, UK).

Jordanous, A. (2014). Stepping back to progress forwards: Setting standards for meta-evaluation of computational creativity. In *Proceedings of 5th International Conference on Computational Creativity*, Ljubljana, Slovenia.

Jordanous, A., & Keller, B. (2012). What makes musical improvisation creative? *Journal of Interdisciplinary Music Studies*, *6*(2), 151–175.

Jordanous, A., & Keller, B. (2016). Modelling creativity: Identifying key components through a corpus-based approach. *PLOS ONE*, *11*(10), e0162959. doi:10.1371/journal.pone.0162959

Kaufman, J. C. (2009). *Creativity 101*. The Psych 101 series. New York: Springer.

León, C., & Gervás, P. (2010). The Role of Evaluation-Driven Rejection in the Successful Exploration of a Conceptual Space of Stories. *Minds and Machines*, *20*(4), 615–634. doi:10.1007/s11023-010-9205-z

Meehan, J. (1981). Tale-Spin. In R. C. Schank & C. K. Riesbeck (Eds.), *Inside computer understanding: Five programs plus miniatures*. Hillside, NJ: Lawrence Erlbaum Associates.

Montfort, N., & Pérez y Pérez, R. (2008). Integrating a plot generator and an automatic narrator to create and tell stories. In *Proceedings of the 5th International Joint Workshop on Computational Creativity*, Madrid.

Pearce, M. T., Meredith, D., & Wiggins, G. A. (2002). Motivations and Methodologies for Automation of the Compositional Process. *Musicae Scientae*, *6*(2), 119–147.

Pease, A., & Colton, S. (2011). Computational Creativity Theory: Inspirations behind the FACE and the IDEA models. In *Proceedings of the 2nd International Conference on Computational Creativity, ICCC-11* (pp. 72–77). Mexico City.

Pease, A., Winterstein, D., & Colton, S. (2001). Evaluating machine creativity. In *Workshop on Creative Systems, 4th International Conference on Case Based Reasoning* (pp. 129–137).

Peinado, F., Francisco, V., Hervás, R., & Gervás, P. (2010). Assessing the Novelty of Computer-Generated Narratives Using Empirical Metrics. *Minds and Machines*, *20*(4), 565–588. doi:10.1007/s11023-010-9209-8

Peinado, F., & Gervás, P. (2006). Evaluation of automatic generation of basic stories. *New Generation Computing*, *24*(3), 289–302.

Pereira, F. C., & Cardoso, A. (2006). Experiments with free concept generation in Divago. *Knowledge-Based Systems*, *19*(7), 459–470. doi:10.1016/j.knosys.2006.04.008

Pérez y Pérez, R., Aguilar, A., & Negrete, S. (2010). The ERI-Designer: A Computer Model for the Arrangement of Furniture. *Minds and Machines*, *20*(4), 533–564.

Pérez y Pérez, R., & Sharples, M. (2004). Three Computer-Based Models of Storytelling: BRUTUS, MINSTREL and MEXICA. *Knowledge-Based Systems*, *17*(1), 15–29.

Plucker, J. A., & Beghetto, R. A. (2004). Why Creativity is Domain General, Why it Looks Domain Specific, and why the Distinction Doesn't Matter. In R. J. Sternberg, E. L. Grigorenko, & J. L. Singer (Eds.), *Creativity: From potential to realization* (Chap. 9, pp. 153–167). Washington, DC: American Psychological Association.

Plucker, J. A., Beghetto, R. A., & Dow, G. T. (2004). Why Isn't Creativity More Important to Educational Psychologists? Potentials, Pitfalls, and Future Directions in Creativity Research. *Educational Psychologist*, *39*(2), 83–96.

Ritchie, G. (2001). Assessing Creativity. In *Proceedings of the AISB symposium on AI and creativity in arts and science* (pp. 3–11). York, UK.

Ritchie, G. (2007). Some Empirical Criteria for Attributing Creativity to a Computer Program. *Minds and Machines*, *17*, 67–99. doi:10.1007/s11023-007-9066-2

Ritchie, G. (2019). The evaluation of creative systems. In T. Veale & F. A. Cardoso (Eds.), *Computational creativity: The philosophy and engineering of autonomously creative systems* (pp. 157–192). Springer.

Sloman, A. (1978). *The computer revolution in philosophy*. Hassocks, UK: Harvester Press.

Tearse, B., Mawhorter, P., Mateas, M., & Wardrip-Fonin, N. (2011). Experimental Results from a Rational Reconstruction of MINSTREL. In *Proceedings of the 2nd International Conference on Computational Creativity, ICCC-11* (pp. 54–59). Mexico City.

Turner, S. R. (1994). *The creative process: a computer model of storytelling and creativity*. Hillside, NJ: Erlbaum.

Whorley, R. P., Wiggins, G. A., & Pearce, M. T. (2007). Systematic evaluation and improvement of statistical models of harmony. In *Proceedings of the 4th International Joint Workshop on Computational Creativity* (pp. 81–88). London.

Whorley, R., Wiggins, G., Rhodes, C., & Pearce, M. (2010). Development of Techniques for the Computational Modelling of Harmony. In *Proceedings of the International Conference on Computational Creativity* (pp. 11–15). Lisbon.

Widmer, G., Flossmann, S., & Grachten, M. (2009). YQX Plays Chopin. *AI Magazine*, *30*(3), 35–48.

Wiggins, G. A. (2006). A preliminary framework for description, analysis and comparison of creative systems. *Knowledge-Based Systems*, *19*(7), 449–458. doi:10.1016/j.knosys.2006.04.009

Chapter 11
Computer-Supported Human Creativity and Human-Supported Computer Creativity in Language

Lorenzo Gatti, Gözde Özbal, Marco Guerini, Oliviero Stock, and Carlo Strapparava

Abstract This chapter is concerned with human–computer collaboration to achieve linguistic creativity. We claim that humans and computers may benefit from each other during the creativity process and we demonstrate concrete examples of systems that allow different degrees of interaction with the user. Then, we focus on computer-supported human creativity, where the computer necessarily requires human intervention, either for providing input or decisions that are essential to the system, or for deciding which outputs are interesting since the computer lacks a quality metric. As examples of systems modeling computer-supported human creativity, we describe GRAPHLAUGH, an interactive system which produces humorous puns by modifying familiar expressions, and SUBVERTISER, a mobile application that allows users to creatively alter the message contained in pictures of posters, billboards, and advertisements. Finally, we focus on human-supported computer creativity, where the burden of the creative process is mainly on the computer, while the human simply mediates the process during key steps whenever required. As an example modeling this type of creativity, we introduce HEADY-LINES, which automatically generates creative headlines combining a well-known expression with a concept from the news.

Lorenzo Gatti
FBK-irst, Trento, Italy.
Current address: University of Twente, The Netherlands. e-mail: l.gatti@utwente.nl

Gözde Özbal
FBK-irst, Trento, Italy. e-mail: gozbalde@gmail.com

Marco Guerini
FBK-irst, Trento, Italy. e-mail: marco.guerini@fbk.eu

Oliviero Stock
FBK-irst, Trento, Italy. e-mail: stock@fbk.eu

Carlo Strapparava
FBK-irst, Trento, Italy. e-mail: strappa@fbk.eu

T. Veale, F. A. Cardoso (eds.), *Computational Creativity*, Computational Synthesis and Creative Systems, https://doi.org/10.1007/978-3-319-43610-4_11

11.1 Introduction

Although creativity is a fundamental attribute of the human species, the increasing capabilities of computer hardware and software are starting to challenge the assumption that creativity cannot exist outside the boundaries of human minds. Currently, human–computer collaboration is commonly used to achieve creativity: many creative systems distill the useful information from available resources that might be too broad for an unaided human mind, while hiding all the inner complexity of the process. Users are typically involved in the creative process, both to compensate for the increased complexity of higher-level creativity and to fully exploit inherent qualities of both humans and computers (Özbal, 2013).

Let us briefly explore human–computer collaboration with reference to the classification of types of creativity provided by Margaret Boden (2009).

Human creativity may benefit from the support of computer systems, especially in the case of what Boden called combinatorial creativity. This type of creativity involves the combination of familiar notions, concepts, and ideas in an unfamiliar way. The juxtaposition of seemingly unrelated concepts, as in an oxymoron, is an example of combinatorial creativity. Creative humans may be helped by a system that assists them, providing a view of the relevant search space at various moments of the creative process. Ideally a system should really understand what these phases are, understand the needs at each moment, and offer an interface that allows humans to have support without being distracted from their mental activity, but, on the contrary, be able to focus better without interference. In theory, one could also think of intelligent suggestions coming from the exploration of fields different from the one under examination.

Exploratory creativity is instead based on the exploration of some well-defined space, where new concepts and ideas can be combined through a generative process. The exploration of the resulting space can also lead both to a better understanding of the limits to which the generative process can be pushed, and to the discovery of regions that have not been considered already.

The third type of creativity in Boden's classification, transformational creativity, is based on alteration of the rules and the geometry of the conceptual space, which can open up the potential for new forms of expression that would not be possible within the rule set of the original space. It does not seem realistic to suggest this type as an opportunity for a system that supports human creativity, perhaps with the exception of offering excellent editing facilities, where human expressivity can easily pass through a cycle of hypothetical creative solutions and fast adjustment of novel ideas.

Another theory that has proved very influential for the field of computational creativity was developed by Fauconnier and Turner (2008). The conceptual blending theory describes one of the basic mechanisms of the creative process, where novel creations are obtained by merging elements and relations that normally belong to different scenarios. The reader can find concrete examples and a focused overview of computational approaches to conceptual blending in other chapters appearing in this book (Martins, Pereira, & Cardoso, 2019; Veale, 2019).

Not only can systems support humans during the creativity process, but it is also possible for a system to utilize the assistance of users, especially for the production of the first two types of creativity (transformational creativity seems to be beyond the limits of automation, at least for the time being). In this case the system conducts the creative process itself, all the way to the output of a specific result. The role of the human is possibly (i) to restrict the search space of the system; (ii) to select those partial solutions that appear most promising or relevant; and (iii) to simply validate the best of the final results, if they cannot be ranked autonomously or if a threshold of quality for human appreciation is impossible to define and to measure against the solution at hand.

The importance of human-supported computer creativity can be well appreciated if one thinks of adaptive creativity, where the goal is to produce different solutions for different sets of people, individuals, or situations. When a vast amount of content needs to be adapted, or when the input (or target of the communication) is changing continuously, it is inconceivable that a human, even if supported in their activity by a system, can produce such solutions. However, a computer can do the job instead, possibly being guided in its search space at key points or receiving feedback on some of the solutions offered.

For linguistic creativity, adaptivity can be very important: consider personalized advertisements, user-adapted humor, irony, and poetry, to mention just a few themes. The potential is simply enormous considering the huge amount of textual content we are exposed to and its constant transformation.

In this chapter, we describe our experience in building human-supported systems that display various forms of linguistic creativity.

We first summarize our early explorations of computational humor, in particular GRAPHLAUGH, a system for the generation of homophonic puns. Then, we introduce SUBVERTISER, a mobile application that allows users to spoof pictures of billboards, replacing their textual content with content automatically modified by a creative system to produce an ironic effect. Finally, we describe HEADY-LINES, a system that takes existing well-known expressions and innovates with them by bringing in a novel concept coming from evolving news. The resulting sentences can be used as creative headlines or adaptive slogans.

In particular, the focus will be on "human-supported computer creativity," i.e., where there is a limited role for human intervention and the burden of the creative process is mainly on the computer, with humans simply mediating the process at key steps.

Computer-supported human creativity focuses on human creativity and computer systems that can facilitate some part of the creative process, for example by providing humans with suggestions.

Human-supported computer creativity is, instead, centered on the creativity of computers, with humans assisting the computer only when necessary, for example by correcting mistakes or by expanding the search space.

11.2 Related Work

Research in creative language generation has thrived in recent years and state-of-the-art computational models of creativity often produce remarkable results, for example those developed by Manurung et al. (2008), Greene, Bodrumlu, and Knight (2010), Guerini, Strapparava, and Stock (2011), Colton, Goodwin, and Veale (2012), to name just a few. Here we introduce some of the systems that have appeared in the literature, focusing in particular on the distinction between noninteractive and interactive systems. For a review of design principles for creativity support tools in well-established areas of industrial interest, the reader may refer to Shneiderman (2007).

11.2.1 Noninteractive Systems

On one side, we have creative systems that require no manual intervention. For example, Lessard and Levison (1992) generate puns consisting of a quoted utterance and an adverb, by finding a configuration of a root word which can be in an adverb form and a sentence semantically linked to this root word (e.g., "'Turn up the heat,' said Tom coldly").

WISCRAIC (Witty Idiomatic Sentence Creation Revealing Ambiguity In Context) creates jokes with the focus on witticisms (i.e., clever and often ironic remarks) based on phonological ambiguity by extracting semantic associations from both the normal context of words and humor-independent lexical entries (McKay, 2002).

The METAPHORISMYBUSINESS[1] "Twitterbot" is a metaphor-generating program that uses the web service described by Veale (2014) to create novel metaphors and publishes them on the Twitter social platform, one every hour.

With all these systems, the user is only "required" to judge, from among the many possible outputs, which ones are particularly interesting.

11.2.2 Minimally Interactive Systems

Other prototypes require a limited amount of user input, either some material to use as a basis for the creative process, or some parameters with which to modify it. This input is typically required during the first steps of the algorithms.

For example, Stock and Strapparava (2003) use semantic field opposition, rhyme, rhythm, and semantic relations to parody an existing acronym, or to generate a new one for a concept provided by the user.

[1] The bot's output is published at https://twitter.com/MetaphorMagnet, and a short description can be found at http://prosecco-network.eu/event/metaphormagnet-creative-metaphor-generating-twitterbot.

STANDUP (Manurung et al., 2008) is a riddle generator that attempts to create a language playground for children with complex communication needs. Children can choose a word from a list, and the system then uses it as the basis for a riddle generated in real time. Despite the positive impact on the children using this system, the overall quality of the jokes has not been evaluated.

Greene, Bodrumlu and Knight (2010) describe a model for poetry generation in which users can control the meter and rhyme scheme. Generation is modeled as a cascade of weighted finite state transducers that only accept strings conforming to the desired rhyming scheme.

Colton, Goodwin and Veale (2012) present a data-driven approach to poetry generation, based on simile transformation. While the user can change some constraints that determine how words are selected (such as their phonetic properties or frequencies), these authors built a system where they "handed over the high-level control," letting daily news influence the mood and theme of the poems. The system also integrates esthetic appreciation mechanisms and provides a commentary on its own work.

In the system developed by Toivanen et al. (2012), the user can specify a topic that is used to generate novel poems by replacing words in existing poetry with morphologically compatible words that are semantically related to the topic.

Possibly closer to a slogan generation system, VALENTINO (Gatti, Guerini, Stock, & Strapparava, 2014) can modify existing textual expressions to obtain more positively or negatively valenced versions. The user provides a sentence and a target score, indicating whether the sentence should become more positive or more negative, and the system then adds, replaces, or deletes words of the original text taking into account their valence, grammatical, and syntactical constraints.

Finally, BRAINSUP (Özbal, Pighin, & Strapparava, 2013), an extensible framework for the generation of creative sentences for educational and advertising applications, lets users force several words to appear in those sentences. BRAINSUP makes heavy use of syntactic information to enforce well-formed sentences and to constraint the search for a solution, and provides an extensible framework into which various forms of linguistic creativity can easily be incorporated.

11.2.3 Interactive Systems

The last type of creative system is the one where the user can intervene during many different steps of the creation process. The computer, in this case, acts as a digital colleague for its human partner (Lubart, 2005), and collaboration between the two is the key to successful creative work.

While this is common in many creative systems outside the linguistic domain, for example in music (Bell & Gabora, 2016), graphics (Davis et al., 2015), and even dance choreography generation (Carlson et al., 2016), it seems that most programs for the generation of text usually allow a lesser degree of interactivity. Still, some examples of truly interactive systems can be found in this domain too.

One such prototype is NAMELETTE (Özbal & Strapparava, 2013), a system for generating creative names. The user can choose a category of products and a set of properties to be underlined. The system finds related concepts and qualities associated with the product, and uses them to generate a name (either a homophonic pun, a metaphor, or a neologism produced by adding a Latin suffix to an English word). During this process, both semantic appropriateness and sound pleasantness of the generated names are taken into account. The user can also intervene in the middle of the generation step, by filtering the concepts and qualities retrieved by the system or adding new ones. As an example of the output of the system, the three homophonic puns generated for an Italian restaurant are *eatalian* (from the combination of "eat" and "Italian"), *pastarant* ("pasta" + "restaurant") and *peatza* ("pizza" + "eat"). As another example, for a cool and sporty brand of sunglasses, NAMELETTE suggests the Latinized names *darkissima*, *polarizium*, and *eyelogia*.

GRAPHLAUGH (Valitutti, Strapparava, & Stock, 2009) (discussed in greater detail in Section 11.3.1) presents an interactive system which generates humorous puns obtained through variation of familiar expressions. The system shows a dynamic graph where the user can choose to start from different familiar expressions, select the replacement words, and tweak other aspects of the generation process while it is still happening.

It is worth noting that, even though fully interactive systems let the user intervene during different stages of the process, they do not necessarily force the user to do so. HEADY-LINES (Gatti, Özbal, Guerini, Stock, & Strapparava, 2015; Gatti, Özbal, Guerini, Stock, & Strapparava, 2016), a system for generating creative headlines for an article, can work in an interactive way, asking the user to select a news article and intervene during the process and select the best headline, but is also capable of working in a fully automatic way, providing a small number of generated headlines for articles in a database and ranking these according to its own output-quality metric. This last prototype, and the distinction between interactive systems that can work both with and without the user, will be the focus of Section 11.4.

11.3 Computer-Supported Human Creativity

In this section, we present a few systems that tackle the problem of supporting humans in the production of creative linguistic content. Such systems are built for different tasks (such as the generation of humorous language or poems), and they support different degrees of interactivity with the user. What characterizes computer-supported creativity is the fact the systems concerned are based on computer tools that may facilitate some phases of the creative production process. They may include subsystems that suggest appropriate words or expressions, systems that may help focus on something and so restrict the creative search space, or, in principle, even tools that automatically assess the value of a certain creative production. Tools of this kind, depending on their cleverness and the quality of the computer–human interface, may be very helpful and ease the production process. Human creativity remains the

main contributor to the result, and that tends to provide a final human-level quality. From an applied point of view, the support of such tools is valuable in cases where one single final output expression is sought. If, instead, the application scenario requires many different productions, possibly in parallel, then it is less attractive, because human creative intervention tends to have a bigger role in the final phase of the production, when different linguistic expressions need to be realized. As we shall see in the next section, this is potentially the case when we take into account adaptivity of the outcome to different audiences if they must be reached at the same time or if we have time constraints, for instance in the case where we want to produce messages that change, continuously, on a short time scale.

11.3.1 GRAPHLAUGH

GRAPHLAUGH was one of the earliest attempts to build an interactive system for producing humorous puns through variation (i.e., word substitution) of familiar expressions (Valitutti et al., 2009). The replacement word is selected according to phonetic similarity and semantic constraints, expressing semantic opposition or evoking ridiculous traits of people.

The system can generate puns such as "Chaste makes waste" (a variation on the proverb "haste makes waste") and "Genital Hospital" (a variation on "General Hospital," a soap opera title).

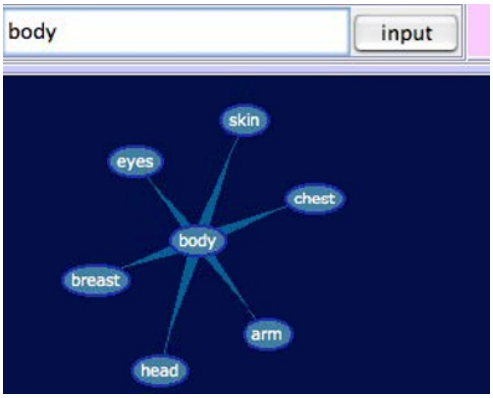
GRAPHLAUGH can automatically generate different types of lexical associations and visualize them in a dynamic graph. Through interaction with nodes and arcs of the network, the user can control the selection of words, semantic associations, and familiar expressions to direct the creative process.

To create its puns, the system requires:

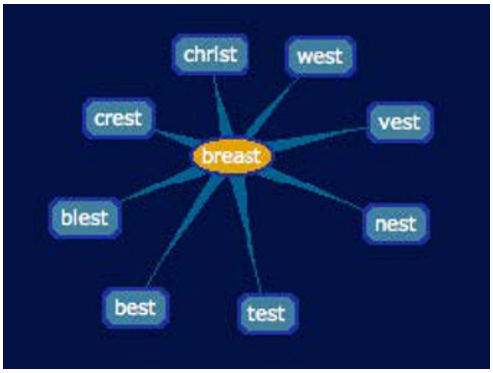
- A list of familiar expressions. A set of 1836 expressions, recognized as "familiar" by English speakers, was collected from the web. It consists of 628 proverbs in common use, 290 famous movie titles, and 918 clichés.
- Phonetic distance. The information on the mapping between words and their phonetic transcription relies on the CMU Pronouncing Dictionary.[2] The algorithm for measuring the phonetic distance is a specific implementation of the Levenshtein distance (Levenshtein, 1966). It is based on a sequence of elementary operations applied to the phonetic expression for a word in order to obtain another word. The weight associated with the substitution operator was modified to take into account the phonetic type, tonic accent, and vowel length.
- Semantic associations. The generation of semantically similar words is based on a measure of lexical similarity. To obtain a vector representation of words, latent semantic analysis (LSA) based on the British National Corpus (BNC)[3] was used. As a measure of semantic similarity between words, GRAPHLAUGH

[2] http://www.speech.cs.cmu.edu/cgi-bin/cmudict

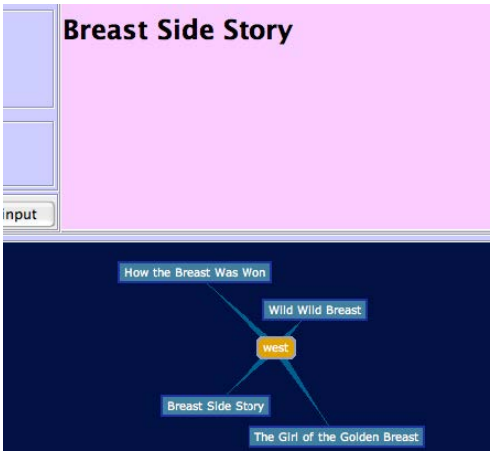
[3] http://www.natcorp.ox.ac.uk/



(a)



(b)



(c)

Fig. 11.1 The GRAPHLAUGH interface: (a) selection of a concept to explore; (b) selection of the new word to be used; (c) selection of the preferred pun.

uses the distance in the resulting vector space, measured as the cosine of the angle between the corresponding vectors. The system also takes into account additional semantic constraints, such as antonymy or semantic domain opposition. These are useful for detecting different types of lexical incongruity. In particular, GRAPHLAUGH uses the *Affective-Weight* function (Strapparava, Valitutti, & Stock, 2006) to assign an affective rating to words, so that the system can combine words with different polarity.

- Dynamic graph. GRAPHLAUGH uses the TouchGraph (Alani, 2003) library to create a dynamic graph that stimulates users to explore a network of concepts and expressions. During the interaction, only the currently selected node and a number of adjacent nodes are visualized. This way, the user is free to explore creative local associations without paying attention to the overall structure.

Initially, the user is required to input a concept from which the creative exploration will start (e.g., "body" in Fig. 11.1a). This concept is expanded using LSA similarity and WordNet relations (such as hyperonymy) to produce a list of candidates that could be used in a pun (e.g., "skin", "arm", "breast", ...). After the user has selected one, a graph showing the words from the familiar expression that can be replaced by the candidate is shown (Fig. 11.1b). Finally, the possible puns constructed are shown and the user can select the best one (Fig. 11.1c) .

GRAPHLAUGH clearly is a system for computer-supported creativity, since every step is determined by the user, from the initial concept selection down to the choice of the pun. The system here is just assisting the human creativity, providing some suggestions in a restricted (and usually promising) part of the whole search space, without the ability to run automatically, since the algorithm cannot decide which alternatives are better.

11.3.2 SUBVERTISER

Another example of computer-supported creativity can be seen in SUBVERTISER, a mobile application that allows users to alter messages on posters, billboards, and advertisements by choosing from among creative suggestions proposed by the system. The rationale behind this "virtual defacement" is to help users fight persuasive advertising in an ironic and nondestructive way.

In a typical usage scenario, users are walking with friends in a city, perhaps shopping or going to see a movie. When they notice a billboard that bothers them, they can start our application and use it to produce a new virtual version of the advertisement with the same layout and visual aspect, but with a creative variation of its existing message.

However, this production mechanism is not entirely automatic. The user has to provide the initial material (i.e. the picture that will be modified), and select a message from a small list of suggestions automatically produced by the system. This

message, when substituted for the original one, is expected to result in a pleasing effect.

(a) (b) (c)

Fig. 11.2 The main steps of SUBVERTISER: (a) text area selection; (b) modified text selection; (c) final output.

Let us briefly describe the algorithm behind SUBVERTISER. When the program starts, the user is asked either to take a picture with the camera in the phone, or to choose a picture from an image gallery. After that, the user needs to identify the region of text to be modified – in our example, "no tea was ever so health friendly" – by moving and resizing a selection rectangle (Fig. 11.2a).

The Tesseract OCR[4] program is then launched on this area, recognizing the text of the message. The OCR program also detects the coordinates of the bounding boxes of every individual word. SUBVERTISER then takes the rectangle containing the first line of text, downscales it to 100 pixels in height, and uploads it to the WhatTheFont[5] font recognition service, using dedicated APIs.

To detect the color of the text, we cluster the pixels of the first line of text into two classes using K-means clustering. The mean of the smaller class is then taken as the color of the text.

Meanwhile, the program also applies the OpenCV "inpainting" algorithm to each line of the original text, to reconstruct the background image that was underneath it. This gives us a clean background where new text can be superimposed.

[4] http://code.google.com/p/tesseract-ocr/

[5] http://www.myfonts.com/WhatTheFont/

After the user has reviewed the message identified by the OCR system to correct any detection errors, this text is sent to a server running VALENTINO (Gatti et al., 2014), a natural language processing system for automatically changing the affect of a short message. For example, given the word "tea," VALENTINO can describe it as "delicious" or "humble," depending on how positive the final output should be. For SUBVERTISER, we ask this system to produce four different valenced sentences, which are then presented to the user, from the most positive to the most negative (Fig. 11.2b).

Once the user has selected one of the computationally created messages ("no hallucinogenic tea was ever so health friendly" in our example), SUBVERTISER determines how to divide the "slanted" text into lines. Since our text modification system not only replaces sentiment-bearing words but also often inserts or removes them, heuristics are used to ensure that the overall text layout is as similar as possible to the original one. The same goes for capitalization, so if a word was capitalized, its replacement will also be capitalized, and the same happens if a new word is inserted into an all-capital chunk.

Then, we use the WhatTheFont service to generate an image of the new text with the same font and color as the original one, and a transparent background.

Finally, this image, with the message selected by the user, is copied inside the bounding boxes of the original text, as detected by the OCR program. The image is shown on screen (Fig. 11.2c) and the user can save it to the image library or share it. More examples of the output of the system are presented in Table 11.1.

Table 11.1 Advertisements modified with VALENTINO

Original text	No tea was ever so health friendly
Slanted text	No hallucinogenic tea was ever so health friendly
Original text	We thought people would want a different kind of car. One that wasn't so much a car.
Slanted text	We thought rude people would want a different kind of car. One that wasn't so much a nice car.
Original text	The manliest low-calories soda in the history of mankind
Slanted text	The manliest low-calories soda in the history of impotent mankind

To sum up, the process of SUBVERTISER requires that the user assists the system in:

1. Providing of a photograph of an interesting advertisement.
2. Indicating which area of the picture should be analyzed by OCR.
3. Correcting the potential mistakes of the automatically recognized text.
4. Selecting a good automatic variation of the original message, as produced by VALENTINO.

While the third point could reasonably be automated without degrading the quality of the results (e.g., by using a spellchecker, especially if combined with a dedicated language model), meaningful results can only be obtained by considering

the interaction between the qualities of the product depicted, the meaning of the text in the advertisement and its modifications, and, possibly, the visual properties of the image. Since SUBVERTISER has no built-in appreciation mechanism for these aspects, it necessarily depends on human intervention and can only work as a "support tool" that suggests potentially useful text modifications.

11.4 Human-Supported Computer Creativity

While creative systems such as those described in the previous section can be very useful for enhancing human creativity, there are scenarios where linguistic creativity is needed (or at least desirable) but the amount of data to be considered in the creative process would make a substantial human contribution infeasible. In these cases, letting the computer guide the process is a necessity.

Two examples of such scenarios come to mind. Personalized advertisements that adapt the message to the user, where the number of recipients would make it impossible to create customized versions manually, could surely benefit from a system that can decide what is the best slogan for any given user. Another case is the creation of catchy headlines, where human intervention can hardly keep up with the flow of freshly produced news. For this particular scenario, a useful tool would be a creative system that can work autonomously – producing multiple headlines for each article and selecting only the best one, so that most articles could get a custom creative title – but that also allows user interaction, so that the most important news items (e.g., those appearing "above the fold" on the front page) can be created "in collaboration" with a copy editor.

This last case could be defined as "human-supported computer creativity": the computer is in charge of the process, and it can produce meaningful results and rank (and, if necessary, discard) them, while still allowing some human interaction during the different steps of the creation process whenever it is needed. For an interactive system to really "master" the creation process, this ranking capability is essential: while SUBVERTISER can often generate humorous results, this is only possible if the user picks a "good" message. The system has no measure of aptness, level of humor or even incongruity with the original message.

A truly human-supported computational system should ideally provide users with a small *ordered* search space where creative results can easily be found. Potentially, it should be able also to produce meaningful results without human intervention, albeit supporting it to further improve the quality of the results. Thus, human-supported computer creativity can also be viewed as a specific subset of mixed-initiative co-creativity (Yannakakis, Liapis, & Alexopoulos, 2014), i.e., the subset in which computational systems can contribute to a large degree to the solution of a problem without performing simple random generation.

In the following section we describe HEADY-LINES, a prototype that shows these characteristics.

11.4.1 HEADY-LINES

HEADY-LINES (Gatti et al., 2015) is a system for the automatic generation of creative headlines that combine a well-known expression with a concept from the news. The system is inspired by catchy headlines such as "The dark side of the sun" (for an article about the dangers of tanning booths) or "This little LED of mine" (for one describing new LED lightbulbs) that often appear in newspapers.

The system is composed of four main modules that deal with (i) retrieving the news of the day from the web, (ii) extracting keywords from the news and expanding them with relevant related concepts, (iii) pairing the news with well-known expressions using state-of-the-art similarity metrics, and (iv) generating a new headline by merging the well-known expression with a keyword from the news, satisfying the lexical and morpho-syntactic constraints enforced by the expression.

On top of this algorithm, we have developed a web interface (Gatti et al., 2016) that hides the technical details from the users (ideally copy editors) and collaborates with them in the creative task of generating a good headline. However, the system can still work in a fully automatic mode, where just a piece of news (or a feed of news) is given as input and the best headline is presented, according to the selection criterion of the algorithm.

Initially, users are presented a list of short descriptions (about 25 words) of the news of the day. From a technical point of view, the news items are retrieved from the RSS feed of BBC News and from The New York Times through its API. Each entry is composed of a headline, a short description of the article, and other metadata, but only the description is used by the system. As they are downloaded, news descriptions are tokenized and part-of-speech tagged using Stanford CoreNLP (Manning et al., 2014). An example of a description provided by our interface is "By any measure, it has been a year from hell for the European Union. And if Britons vote to leave the bloc, next year could be worse." (from The New York Times).

Once the user has selected an interesting news event from the list, its description is presented in a new page with its key concepts highlighted (Fig. 11.3). In particular, stop words and irrelevant words fade to gray, while the defining elements for that news are colored differently, depending on their category. At the moment we are differentiating betwen (i) named entities, (ii) important keywords for which we have some knowledge, and (iii) important but unrecognized keywords. Users are also presented with an additional set of related concepts which are derived from the important words recognized in the sentence. The user can remove any of the identified keywords or related concepts from the list just by clicking on them, or click on a word deemed irrelevant to change its status.

We define the importance of each word as the number of times that word appears in a news corpus (23,415 news documents from the LDC GigaWord corpus (Parker, Graff, Kong, Chen, & Maeda, 2011)), divided by the total number of headlines occurring in the corpus (i.e., the probability of the word). Words under a certain threshold are considered as "key concepts." The "related keywords" that the users see are simply synonyms and derivationally related forms of these words, obtained from WordNet. The named entities are detected by utilizing CoreNLP. In the example

above, the system identifies *hell*, *European Union*, *Britons*, and *bloc* as key terms. It expands this list by retrieving concepts such as the noun *Brit* (a synonym of *Britons*) and the adjective *infernal* (from the noun *hell*).

A list of well-known expressions is then presented to the user, with the expressions that are most related to the news appearing at the top. The user can disable some of the entries by clicking on them (Fig. 11.4) The relatedness is calculated using a skip-gram model (Mikolov, Sutskever, Chen, Corrado, & Dean, 2013) trained on the words of the GigaWord corpus. To compare the news with each expression, we construct a vector representation of the former by summing the vectors of its keywords. Similarly, we build a vector representation of each expression based on its words (after removing the stop words). The expressions that do not reach a certain similarity threshold are discarded. This ensures at least a minimum degree of relatedness between the news and the well-known expression. For the above example, the most similar well-known expression is the national anthem God Save the Queen, followed by the song Son of a Preacher Man. The similarity of the sentences is due to *queen* being related to *Britons*, while *God* and *preacher* are related to *hell*.

The list of the new potential headlines is then shown to the user. They are ranked from best to worst, but the user can click on any of the sentences and see a final page with the headline, along with the starting description, to see how fit it is for the news. The sentences are modified by taking into account the lexical and syntactic constraints imposed by the original expression. This is accomplished by using a database of tuples that stores, for each relation in the dependency treebank of the LDC GigaWord corpus, its occurrences with specific "governors" (heads) and "dependents" (modifiers), similarly to the approach of Özbal et al. (2013). For each lemma *w* in a well-known expression, we determine all the words that are connected to *w* by a dependency relation. Then, we calculate how likely it is that each keyword *k* from the news articles that passed the similarity filter can replace a *w* that is the same part of speech. We can then select the slot containing the word *w* to be replaced, and the best keyword *k* for each news article, by simply maximizing this dependency likelihood. In this case a threshold is enforced also, so that sentences that do not reach a satisfactory level of grammaticality are removed. Finally, the morphology of

Fig. 11.3 Key concepts of the news, and words related to them.

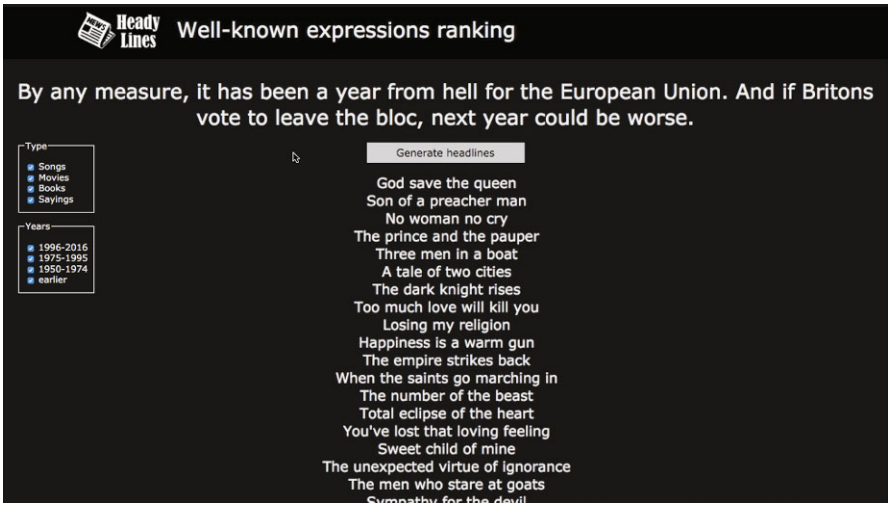


Fig. 11.4 Well-known expressions sorted by similarity to the news.

the replaced word *w* is applied to *k* by using MorphoPro (Pianta, Girardi, & Zanoli, 2008) and the modified sentence is generated. To rank the final output, the system sorts each modified sentence according to its mean rank with respect to similarity and dependency scores, thus balancing the scores for grammaticality and relatedness to the news. The lower the mean, the better the system considers the headline. For our example, the system will offer a very appropriate "God save the bloc" as the best headline. More examples of the output of the system can be seen in Table 11.2.

As we have seen, HEADY-LINES is a system that models human-supported computer creativity. It can automatically produce meaningful and creative content, discarding solutions that are not "convincing" enough according to its own output-quality metric. Nevertheless, the user is allowed to use the system interactively to enhance the output quality.

11.5 Conclusions

The focus of this chapter was the distinction between computer-supported human creativity and human-supported computer creativity. Both are concerned with the production of creative artifacts, but they assign different roles to computers and humans.

In the first case, the computer facilitates part of the creative process, for example by providing humans with suggestions. The system, however, is not entirely autonomous, and human creativity is still key to a successful production.

In the second case, instead, computers can produce creative artifacts on their own, but with the potential for users to intervene during the creative process to enhance

Table 11.2 Output examples

Description	... it has been a year from hell for the European Union. And if Britons vote to leave the bloc, next year could be worse.
Expression	God save the Queen
Headline	God save the bloc
Description	Scientists have reconstructed how an ancient reptile swam in the oceans at the time of the dinosaurs.
Expression	The number of the beast
Headline	The ocean of the beast
Description	WTO finally reached deals, capping a ministerial conference ... where rich and poor countries had been split over the path of trade reforms.
Expression	Bridge over troubled water
Headline	Bridge over troubled division
Description	... Australia captain Steve Smith has passed most of the tests presented in leading a team in the throes of transition.
Expression	The empire strikes back
Headline	The captain strikes back
Description	Martin Shkreli resigns as chief executive of Turing Pharmaceuticals following his arrest on Thursday on securities fraud charges.
Expression	Crime and punishment
Headline	Fraud and punishment

the output, for example by correcting the mistakes of the system or by expanding its search space.

We have shown prototypes of systems of both types, and argue that both can be useful in applied scenarios, the former as a creativity-enhancing tool, the latter when a vast amount of content needs to be produced, or if there is a need to adapt a message to a large number of users, or when the inputs are changing continuously, such as when one is producing news or advertisements for the web.

References

Alani, H. (2003). TGVizTab: An ontology visualisation extension for Protégé. In *Proceedings of Knowledge Capture, Workshop on Visualization Information in Knowledge Engineering*.

Bell, S., & Gabora, L. (2016). A music-generating system based on network theory. In *Proceedings of the 7th International Conference on Computational Creativity*.

Boden, M. A. (2009). Computer models of creativity. *AI Magazine*, *30*(3), 23.

Carlson, K., Pasquier, P., Tsang, H. H., Phillips, J., Schiphorst, T., & Calvert, T. (2016). Cochoreo: A generative feature in idanceForms for creating novel keyframe animation for choreography. In *Proceedings of the 7th International Conference on Computational Creativity* (pp. 380–387).

Colton, S., Goodwin, J., & Veale, T. (2012). Full-FACE poetry generation. In *Proceedings of the 3rd International Conference on Computational Creativity* (pp. 95–102).

Davis, N., Hsiao, C.-P., Singh, K. Y., Li, L., Moningi, S., & Magerko, B. (2015). Drawing apprentice: An enactive co-creative agent for artistic collaboration. In *Proceedings of the 2015 ACM SIGCHI Conference on Creativity and Cognition* (pp. 185–186). ACM.

Fauconnier, G., & Turner, M. (2008). *The way we think: Conceptual blending and the mind's hidden complexities*. Basic Books.

Gatti, L., Guerini, M., Stock, O., & Strapparava, C. (2014). Sentiment variations in text for persuasion technology. In *Proceedings of the 9th International Conference on Persuasive Technology* (pp. 106–117).

Gatti, L., Özbal, G., Guerini, M., Stock, O., & Strapparava, C. (2015). Slogans are not forever: Adapting linguistic expressions to the news. In *Proceedings of the 24th international conference on artificial intelligence* (pp. 2452–2458).

Gatti, L., Özbal, G., Guerini, M., Stock, O., & Strapparava, C. (2016). Heady-Lines: A creative generator of newspaper headlines. In *Companion publication of the 2016 International Conference on Intelligent User Interfaces* (pp. 79–83).

Greene, E., Bodrumlu, T., & Knight, K. (2010). Automatic analysis of rhythmic poetry with applications to generation and translation. In *Proceedings of the 15th Conference on Empirical Methods in Natural Language* (pp. 524–533).

Guerini, M., Strapparava, C., & Stock, O. (2011). Slanting existing text with Valentino. In *Proceedings of the 16th International Conference on Intelligent User Interfaces* (pp. 439–440).

Lessard, G., & Levison, M. (1992). Computational modelling of linguistic humour: Tom Swifties. In *Proceedings of the 1992 Joint Annual Conference of the Association for Computers and the Humanities and the Association for Literary and Linguistic Computing* (pp. 175–178).

Levenshtein, V. I. (1966). Binary codes capable of correcting deletions, insertions, and reversals. *Soviet Physics Doklady*, *10*(8), 707–710.

Lubart, T. (2005). How can computers be partners in the creative process: Classification and commentary on the special issue. *International Journal of Human–Computer Studies*, *63*(4–5), 365–369.

Manning, C. D., Surdeanu, M., Bauer, J., Finkel, J., Bethard, S. J., & McClosky, D. (2014). The Stanford CoreNLP natural language processing toolkit. In *Proceedings of the 52nd Annual Meeting of the Association for Computational Linguistics* (pp. 55–60).

Manurung, R., Ritchie, G., Pain, H., Waller, A., O'Mara, D., & Black, R. (2008). The construction of a pun generator for language skills development. *Applied Artificial Intelligence*, *22*(9), 841–869.

Martins, P., Pereira, F. C., & Cardoso, F. A. (2019). The nuts and bolts of conceptual blending: Multi-domain concept creation with Divago. In T. Veale & F. A. Cardoso (Eds.), *Computational creativity: The philosophy and engineering of autonomously creative systems* (pp. 91–118). Springer.

McKay, J. (2002). Generation of idiom-based witticisms to aid second language learning. In *Proceedings of the Twente Workshop on Language Technology 20* (pp. 70–74).

Mikolov, T., Sutskever, I., Chen, K., Corrado, G. S., & Dean, J. (2013). Distributed representations of words and phrases and their compositionality. In *Proceedings of the 26th Conference on Advances in Neural Information Processing Systems* (pp. 3111–3119).

Özbal, G. (2013). *Computational approaches to linguistic creativity for real world applications* (Doctoral dissertation, University of Trento).

Özbal, G., Pighin, D., & Strapparava, C. (2013). BRAINSUP: Brainstorming support for creative sentence generation. In *Proceedings of the 51st Annual Meeting of the Association for Computational Linguistics* (pp. 1446–1455).

Özbal, G., & Strapparava, C. (2013). Namelette: A tasteful supporter for creative naming. In *Companion publication of the 2013 International Conference on Intelligent User Interfaces* (pp. 55–56).

Parker, R., Graff, D., Kong, J., Chen, K., & Maeda, K. (2011). English GigaWord, 5th edn, ldc2011t07. *Linguistic Data Consortium*.

Pianta, E., Girardi, C., & Zanoli, R. (2008). The TextPro tool suite. In *Proceedings of the 6th International Conference on Language Resources and Evaluation* (pp. 2603–2607).

Shneiderman, B. (2007). Creativity support tools: Accelerating discovery and innovation. *Communications of the ACM*, *50*(12), 20–32.

Stock, O., & Strapparava, C. (2003). Getting serious about the development of computational humor. In *Proceedings of the 18th International Joint Conference on Artificial Intelligence* (Vol. 3, pp. 59–64).

Strapparava, C., Valitutti, A., & Stock, O. (2006). The affective weight of lexicon. In *Proceedings of the 5th International Conference on Language Resources and Evaluation* (pp. 423–426).

Toivanen, J. M., Toivonen, H., Valitutti, A., & Gross, O. (2012). Corpus-based generation of content and form in poetry. In *Proceedings of the 3rd International Conference on Computational Creativity* (pp. 175–179).

Valitutti, A., Strapparava, C., & Stock, O. (2009). GraphLaugh: A tool for the interactive generation of humorous puns. In *Proceedings of the 3rd Conference on Affective Computing and Intelligent Interaction* (pp. 634–635).

Veale, T. (2014). A service-oriented architecture for metaphor processing. In *Proceedings of the 2nd workshop on metaphor in NLP* (pp. 52–60).

Veale, T. (2019). From conceptual mash-ups to bad-ass blends: A robust computational model of conceptual blending. In T. Veale & F. A. Cardoso (Eds.), *Computational creativity: The philosophy and engineering of autonomously creative systems* (pp. 71–89). Springer.

Yannakakis, G. N., Liapis, A., & Alexopoulos, C. (2014). Mixed-initiative co-creativity. In *Proceedings of the 9th Conference on the Foundations of Digital Games*.

Chapter 12
Representing Social Common-Sense Knowledge in MEXICA

Rafael Pérez y Pérez

Abstract One of the most difficult tasks in plot generation is the production of sequences of actions that make sense within the context of a narrative. Previous systems have employed predefined story structures, or variations of them, to guarantee the generation of coherent outputs. However, this method may produce rigid and predictable tales. Thus, it is necessary to find alternatives that allow more flexible stories to be composed. The use of common-sense knowledge (CSK) that includes social features is a useful tool in this endeavour. This chapter describes how MEXICA, a plot generator, exploits such social CSK to generate narratives avoiding the use of predefined story structures. The results suggest that this is a promising approach.

12.1 Introduction

I define computational creativity as the interdisciplinary study of a creative process employing computers as the core tool for reflection and generation of novel knowledge (Pérez y Pérez, 2015a). Thus, its main purpose is to contribute to the understanding of this phenomenon. Plot generation is among the most interesting and challenging topics of study in the area. And one of its most demanding goals is to build agents capable of generating sequences of actions that make sense. This task is so complicated that, in the past, scientists have faced this challenge by using predefined story structures or characters' explicit goals to guarantee the coherence of the tale. As a result, automatic storytellers may produce logical narratives that are rigid and predictable. This limitation has already been pointed out:

> We shall introduce the term *story-predictability* to indicate the degree to which the output of a computerised-storyteller can be predicted when the content of the system's knowledge-structures are known... we propose that the output's predictability depends strongly on the

Rafael Pérez y Pérez
Universidad Autónoma Metropolitana, Cuajimalpa, México.
e-mail: rperez@correo.cua.uam.mx

T. Veale, F. A. Cardoso (eds.), *Computational Creativity*, Computational Synthesis and Creative Systems, https://doi.org/10.1007/978-3-319-43610-4_12

> amount of predefined knowledge structures... We suggest that if the story-predictability is low, the system is more likely to be evaluated as representing a creative process. (Pérez y Pérez & Sharples, 2004)

Even now that we have access to endless amounts of information that allow these kinds of system to produce thousands of combinations employing the same narrative structure, their contribution to the understanding of the creative process is still limited. An alternative to overcome this shortcoming is the study of how to represent common-sense knowledge (CSK) in computer models of plot generation that allows more flexible and interesting storytellers to be produced. That is what this chapter is about.

The following lines elaborate this idea. 'Most scholars now see narrative... and a host of rhetorical figures not as 'devices' for structuring or decorating extraordinary texts but instead as fundamental social and cognitive tools' (Eubanks, 2004). In this way, storytelling plays a fundamental role in our ability to make sense of our cultural environment and therefore to understand the world. Some authors describe stories as sequences of events (e.g. Bremond (1996)). Thus, an important research question is: how do we associate ideas in a way that allows a plot to progress? A narrative must fulfil several constraints:

> To be acceptable as a tellable story, a piece of text should satisfy certain minimum requirements. It must introduce characters and setting and advance one or more key characters through activities within the setting to produce a plot. It must show a coherence and narrative flow between activities and settings over time. It should display an overall integrity and closure, for example with a problem posed in an early part of the text being resolved by the conclusion. It should generate suspense by setting up difficulties faced by the characters or tension in the plot. It should display originality (Pérez y Pérez & Sharples, 2004)

For the purpose of this chapter I would like to develop one of these characteristics: 'advance one or more key characters through activities within the setting to produce a plot'. The reader can picture a story that starts by describing a couple who have just woken up in a cabin in the middle of the forest that is covered by snow. How would an author continue this tale? Perhaps the couple would walk in the nearby area looking for wood in order to warm the cabin; they might encounter a hungry wolf; and so on. These options seem sensible. However, this hypothetical author would probably never imagine a scene where the two characters are wearing swimsuits or they meet a giraffe in the surrounding area. These alternatives do not *make sense*. The question here is how does this associative process of plausible ideas work? How do we know that a situation is suitable for the progress of a narrative? What does it mean 'to make sense'? The question of how to develop a coherent chain of events that fits the requirements of a story is central to the understating of creative writing; however, we are far from knowing the answer. Computer models of plot generation must contribute to this endeavour. This chapter describes how computer representations of social CSK might help us to advance towards this goal.

Because this book is intended to introduce readers to the fascinating world of computational creativity, rather than describing a complex model of writing I have decided to present a gentle approach to how I use CSK in my computerised storyteller. Thus, in the following lines I reflect on some of the main characteristics of CSK found

in the literature; then, I analyse the features of the main building blocks of any story, namely actions; next, I provide a general explanation of the solution I implemented in my computer model MEXICA; and, finally, I discuss the consequences of this approach, its limitations and its opportunities.

12.2 Common-Sense Knowledge

Common-sense knowledge is an attractive area of study that has attracted interest from researchers in fields including psychology, artificial intelligence and cognitive science. The literature offers several definitions, for instance: 'Common-sense knowledge includes the basic facts about events (including actions) and their effects, facts about knowledge and how it is obtained, facts about beliefs and desires. It also includes the basic facts about material objects and their properties' (McCarthy, 1989) and 'Common sense knowledge can be viewed as a collection of simple facts about people and everyday life, such as "Things fall down, not up", and "People eat breakfast in the morning"' (Lieberman, 2008).

> [It] is knowledge about the everyday world that is possessed by most people in a given culture – what is widely called 'common sense knowledge'. While 'common sense' to the ordinary people is related to 'good judgment' as a synonymous, the Artificial Intelligence community uses the term 'common sense' to refer to the millions of basic facts and understandings that most people have. For example, the lemon is sour; to open a door, you must usually first turn the doorknob; if you forget someone's birthday, they may be unhappy with you. (Anacleto et al., 2006)

Although useful, these descriptions do not seem to capture all the associated complexity. For example, it is necessary to consider that:

- Much CSK depends on social and cultural traditions. So, it might change in time and space.
- Much CSK is contextual. An action that makes sense in a given context might be inexplicable when the context changes.
- CSK depends on one's experience. For example, if a person who grew up in a city is suddenly abandoned in the middle of the Amazon, that person probably will not survive because of their lack of basic understanding of the surroundings.
- What is considered as specialised knowledge by one group of individuals might be considered as common-sense knowledge by a different group. In the above example about the Amazon, a basic understanding of the surroundings might be categorised as simple everyday CSK by the locals. Transforming newly acquired knowledge into CSK might be part of understanding a culture, learning how to use a new technology, becoming an expert in a specific domain and so on.

Thus, CSK in artificial intelligence is not just a problem about collecting thousands of basic facts about everyday life and a problem about how to administrate such an amount of information. It also is related to developing adequate knowledge representations and computer processes that allow one to implement systems that

show the required flexibility. For this reason, it is important to capture at least part of this complexity; otherwise, we will always develop limited systems.

Computer models of plot generation provide an excellent framework that allows the study of CSK. Although they are far from capturing the difficulty involved in the generation of human narratives, these types of system are rich sources of information. Story generators have been strongly focused on the content level (what happens in the story) rather than on the level of expression (how the story is told). For this reason, they can be characterised more precisely as plot generators (Montfort & Pérez y Pérez, 2008). In this chapter I concentrate on the content level. I am interested in studying the following aspects:

- What type of CSK is explicitly represented in a system (and therefore what type the program can manipulate), what type of CSK is hardwired and what type of CSK is not represented at all.
- How a system makes use of CSK during plot generation to produce sequences of actions that make sense within the narrative .
- How easy it is to update CSK within a system.

I employ MEXICA to illustrate all these points.

12.3 Characteristics of Story Actions

Claude Bremond (1966/1996) describes a story as a sequence of actions. In most computer models of writing, story actions include preconditions, postconditions and one or more characters. The incidents in the tale can have different *levels of description*, i.e. they can express a happening employing more or less detail. For example, the following three examples describe a killing:

John drew his silver gun and slowly pulled the trigger and killed the villain.

John shot his gun and killed the villain.

John killed the villain.

Each event has a different level of description associated with it: the first line is classified as *rich*, the second as *moderated* and the third example as *reduced*. In the three cases, the reader must employ his CSK to put together the missing information. However, actions with reduced descriptions demand the construction of more information than those with rich explanations. In this way, computer models of plot generation that use reduced descriptions translate an important part of the process of building the story world into the reader. Because we do not understand yet how narrative generation works in humans, this seems an adequate tactic for computer-based plot generators.

In general, preconditions and postconditions of story actions are employed to codify CSK, although they may also be used for other purposes (e.g. to force specific situations to arise during the generation of a narrative; see an example later). In this

chapter I divide such knowledge into two categories: social and logical (Pérez y Pérez, 2015a). *Social CSK* depends on social or cultural contexts. For example, the action where character A shouts at character B might have different consequences in narratives that describe different folk traditions. *Logical CSK* is independent of such customs. It describes rules about how the physics of the story world works. For example, if character A wants to walk into a room the door must be open; character C must be ill in order for character B to heal character C; and so on. It is clear that, in fiction, the characteristics of the story world are defined by the creator. So, the rules describing such features might change from tale to tale or from author to author. But once they are established for a particular work, they should not be altered. Thus, the preconditions and postconditions include social and logical CSK.

The main role of preconditions is to guarantee that an action performed makes sense within the narrative. When the level of description of such an action is high, its set of preconditions is more elaborate. For example, the deed 'John drew his silver gun and slowly pulled the trigger and killed the villain' might perhaps have the following requirements:

Logical:
John and the villain are situated in the same location.
John possesses a gun.
The gun is silver.
John is wearing a holster.
Social:
John hates the villain.

The prerequisites for performing 'John killed the villain' might perhaps be:

Logical:
John and the villain are situated in the same location.
Social:
John hates the villain.

Some computer-based plot generators employ only the minimum number of preconditions required to ensure that an action makes sense within a story. We refer to these as *open*. However, preconditions may also be used by designers to force the system to produce 'interesting' situations in the narrative. For example, we can include as part of the requirements to perform the action 'kill' that the villain is an old person, or that John and the villain are brothers. We can even go to the extreme of requiring the following facts: John and the villain are brothers; John is the town's sheriff; the villain is a handsome man; and the villain stole John's woman. In this last example, the preconditions are so constrained and elaborate that it is difficult to imagine how this action could be used in other type of stories. We refer to these as *closed*. Thus, we can imagine a continuum, known as *granularity*, with two poles:

- Open preconditions (open-P). These are general enough that they allow actions to be performed in a variety of circumstances but at the same time guarantee coherence between deeds.
- Closed preconditions (closed-P). These require that very singular and elaborate circumstances take place in the story before the action can be performed. As a consequence, such an action can only be completed in very strict situations. They might be employed to guarantee the development of suspense, drama and so on within the narrative.

The position in the continuum depends on the number and characteristics of the requirements; typically closed-P has more elements than open-P. There is another perspective from which preconditions can be analysed. They can represent physical objects (e.g. Cinderella's shoe), locations (e.g. a palace), characters' features (e.g. john has a spot on his right cheek, John is tall), relations between characters (e.g. John is the brother of Paul, John is in love with Laura) and so on. Thus, it is possible to represent things ranging from very *concrete preconditions* (concrete-P) (e.g. the object 'the blue ball' or the feature 'John has a small spot on his right cheek') to *intangible preconditions* (intangible-P) (e.g. being in love with, being evil). I refer to this continuum as the *level of abstraction*. Table 1 shows several different possible preconditions for the action where John kills the villain.

Table 12.1 Possible preconditions for the action where John kills the villain

		LEVEL OF ABSTRACTION	
		Concrete-P	**Intangible-P**
GRANULARITY	**Closed-P**	John possesses a gun. The gun is silver. John is wearing an old holster. The villain has blue eyes.	John hates the villain. The villain is in love with John's wife. The wife is a friend of the villain's young brother. The young brother hates John.
	Open-P	John possesses a gun.	John hates the villain.

Finally, postconditions specify how the story world is modified; in this way, *their main role is to progress the narrative*. As in the previous case, they can be logical or social, and they have associated with them a granularity (that goes from open-T to closed-T) and a level of abstraction (that goes from concrete-T to intangible-T).

From this analysis I suggest that:

- Social CSK must have a strong influence during the unravelling of a narrative.
- A narrative generator that works at the level of expression (how the story is told) would probably benefit from using preconditions and postconditions at a concrete level of abstraction; a system working at the content level (what happens in the story) would benefit from using an intangible level of abstraction.

- A system that uses predefined story structures usually employs closed-P, while one that attempts to avoid such types of structure tends to use open-P.
- Actions with a reduced level of description can easily be employed in different circumstances. For example, 'character A killed character B' might be included in a story about Cleopatra, Cuauhtémoc, Martin Luther King or the riots in London in 2011. By contrast, those with a high level of description are more constrained and therefore their requirements are harder to satisfy.
- Open-P and open-T tend to be intangible, while closed-P and closed-T tend to be concrete (although, of course, it is possible to mix these two categories).

Thus, there seems to be a correlation between level of description of an action and the granularity and level of abstraction of its preconditions. Actions with a rich level of description have a tendency to employ closed-P and concrete-P, while actions with a reduced level of description have an inclination to use open-P and intangible-P. In the same way, flexible representations of social CSK seem to be associated with the use of open and intangible preconditions and postconditions (see the examples in the next sections).

12.4 MEXICA

MEXICA (Pérez y Pérez, 1999) is based on the engagement–reflection cognitive account of writing (Sharples, 1999). During engagement, the system generates sequences of actions. During reflection, the system evaluates the material generated so far and, if necessary, modifies it. Then, the program switches back to engagement and the cycle starts again until the story is finished.

This text focuses on describing the general aspects of MEXICA's CSK representation. Readers interested in a general description of how the whole system works are referred to (Pérez y Pérez, 2015b, 2015c); those interested in details of the engagement–reflection computer model of creativity and the system's architecture are referred to (Pérez y Pérez, 1999, 2007; Pérez y Pérez & Sharples, 2001); those interested in comparing different computerised storytellers are referred to (Gervás, 2009; Pérez y Pérez & Sharples, 2004).

MEXICA works as follows. The user provides what are known as the dictionary of story actions and the set of previous stories. The dictionary includes the names of all actions that can be performed by characters, and their preconditions and postconditions. The previous stories are examples of well-formed narratives and represent the experience of the agent; they are written in a rigid format designed to make it easier to process this information (the program does not work with natural language). Employing these two files, the system builds its knowledge base. Then, through engagement–reflection cycles, MEXICA generates coherent, novel and interesting narratives (Fig. 12.1). The following lines provide some details. Each time an action is performed, the narrative's context, i.e. the state of affairs in the story world, is modified. MEXICA employs the current context as a cue to probe

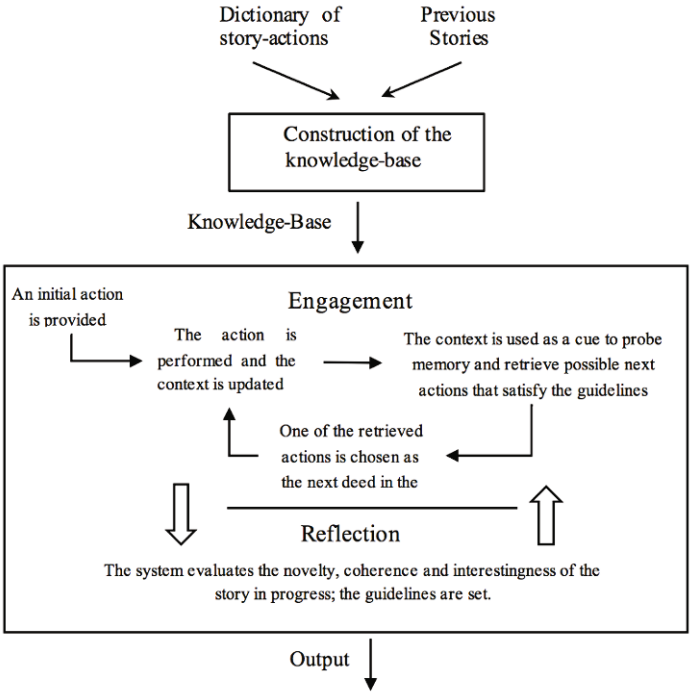


Fig. 12.1 General description of how MEXICA works.

its memory and retrieve compelling deeds to progress the story. One of these deeds is selected at random as the next event in the tale; then, the system performs the chosen action, the context is updated and new options to continue the narrative are retrieved. This cycle, known as engagement, is repeated several times. Then, the system stops and switches to reflection in order to evaluate whether the material produced so far is interesting (i.e. if it has a structure representing a development, climax and resolution) and novel (i.e. if it is not similar to any of the previous stories); if necessary, the agent modifies the plot in progress to guarantee its coherence (e.g. new events might be inserted to fully explain or justify a particular situation that emerged during engagement). The evaluation process adjusts the values of a group of variables known as the guidelines, whose function is to constrain the production of material during engagement. Then, the agent switches back to engagement and the cycle continues until the story is finished. In this way, avoiding the use of predefined story structures, the system continuously progresses the narrative and then evaluates the work in progress (compare the processes performed during reflection with the ideas on evaluation described by Jordanous (2019), (Gervás, 2019) and (Ritchie, 2019) in this volume).

MEXICA works on the assumption that CSK can be represented in terms of emotional links and tensions between characters. Examples of emotional links are

character A is in love with character B, character A hates character B, and so on. The system includes several types of emotional links, which are implemented in discrete terms with a value in the range from −3 to +3. However, for reasons of clarity, in this chapter we describe only two of them: type 1 represents a continuum between brotherly love and hate and is represented by a solid arrow joining two characters, and type 2 represents a continuum between amorous love and feeling hatred and is represented by a dashed arrow joining two characters (Fig. 12.2). In MEXICA, a tension is triggered when the health of a character is at risk (represented by the mnemonic Hr), when the life of a character is at risk (represented by the mnemonic Lr), when a character is made a prisoner (represented by the mnemonic Pr) and when a character dies (represented by the mnemonic Ad). Emotional links and tensions between characters are activated by the postconditions of the actions. There exists a second kind of tension, called *inferred*, which is automatically triggered when the system detects that one character hates a second one and both are located in the same place (known as potential danger and represented by the mnemonic Pd), two characters are in love with a third character (known as love competition and represented by the mnemonic Lc), or a character establishes opposite emotional links towards another character, i.e. when a character A both hates and loves character B (known as clashing emotions and represented by the mnemonic Ce). Tensions between characters are represented by jagged arrows joining the two characters (Fig. 12.2). The number and type of emotional links and tensions between characters are hardwired in the system.

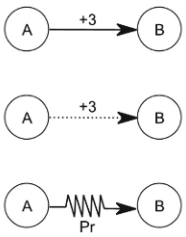


Fig. 12.2 Representations of emotional and tensional links between characters.

MEXICA is based on story actions. All actions have associated with them a set of preconditions and postconditions in terms of emotional links and tensions. In MEXICA, tensions are associated with logical CSK; emotional links and inferred tensions are associated with social CSK. For example, the action character A heals character B requires that character B is ill or injured in order for it to be performed (i.e. a tension of type health at risk). Otherwise, it does not make sense to cure character B. This precondition does not depend on the social context. On the other hand, the action where character A insults character B might have different requirements depending on the social context. In some cultures is required only that character A dislikes B (an emotional link of intensity −1 and type 1 developed from A to B), while in other cultures it requires that character A hates character B (an emotional link of intensity

−3 and type 1 developed from A to B). The postconditions of actions work in similar way. The consequence of the action character A heals character B is that character B is healthy again (the tension health at risk is deactivated). It does not depend on cultural traditions. However, the consequence of the action A shouts at B depends on the social context: B might feel uncomfortable or B might get really irritated. It is also possible to represent relations between friends who do not participate in an action. For instance, if character A saves the life of character B, not only character B but also all her friends will be grateful to character A. In this way, the user of MEXICA can determine the preconditions and postconditions for each action based on the type of CSK that the user wants to represent. This characteristic provides the system with great flexibility and allows experimenting in different scenarios.

In MEXICA, a story is represented in two ways: (1) as a sequence of actions and (2) as a group of emotional links and tensions between characters that progress over the time of the story (for details see (Pérez y Pérez, 2007)). Such a group is known as the story context. So, each time an action is performed, its postconditions are triggered and the story context is updated. Figure 12.3 shows a sequence of six actions and their corresponding context. Each character is represented by a circle: K stands for Jaguar knight, P for the princess and E for the enemy. The location of each character is indicated in the figure. So, at time = 1 action 1 is performed, generating the story context 1; at time = 2, action 2 is performed, generating context 2, and so on. For the example in Fig. 12.3, at time = 1 the action Jaguar Knight had an accident is performed; so, the system updates the story context, triggering the tension health at risk (Hr) of Jaguar Knight (this is the postcondition of action 1). At time = 2, the action princess cured Jaguar Knight is performed; the postconditions of the action indicate that the tension Hr must be deactivated (the knight is cured) and that the knight is very grateful towards the princess (an emotional link of type 1 and intensity +3). At time = 3, the action enemy kidnapped the princess is performed, which produces several consequences: the princess hates the enemy (emotional link of type 1 and intensity −3); because the knight is very grateful to the princess (they are friends), he also hates the enemy; because the enemy has kidnapped the princess, the tension prisoner (Pr) is triggered; by default, when a character kidnaps other character, both are located in the forest; because the princess hates the enemy and both are in the same location, the tension potential danger (Pd) is triggered. At time = 4, the action Jaguar Knight looked for and found the princess is performed; as a consequence, the three characters are located in the same place; because the knight hates the enemy and now both are in the same location, a tension of potential danger from the knight towards the enemy is triggered. At time = 5, the enemy runs away; therefore, the enemy changes his location and the tensions of potential danger are automatically deactivated. Finally, at time = 6, the knight rescues the princess; as a result, the tension Pr is deactivated, the princess is very grateful towards the knight (an emotional link of type 2 and intensity +3) and the knight is very happy with himself (an emotional link of type 1 and intensity +2 towards himself).

The story context at time = n represents the core events in the story in terms of emotional links and tensions. For example, in Fig. 12.3 at time = 4 it is possible to trace the core incidents that have occurred from time = 1 to time = 4: the knight is very

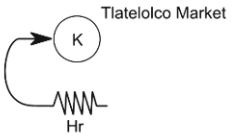
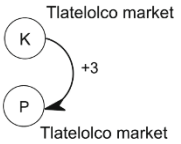
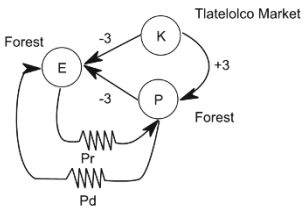
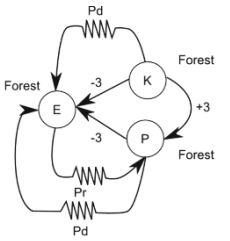
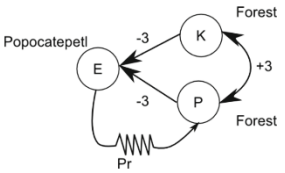
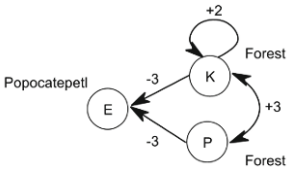
Time	Action	Story Context
1	Jaguar Knight had an accident	
2	Princess cured Jaguar Knight	
3	Enemy kidnapped the princess	
4	Jaguar Knight looked for and found the princess	
5	Enemy ran away	
6	Jaguar Knight rescued the princess	

Fig. 12.3 The progression of a story.

grateful to the princess because she cured him; the princess and the knight hate the enemy because he kidnapped the princess; as a consequence of the kidnapping, the princess is a prisoner; because the three characters are situated in the same location (the knight looked for and found the enemy), there are two tensions of potential

danger coming from the princess and the knight towards the enemy (they hate the enemy). *Thus, story contexts record the state of affairs of the story world, in terms of emotional links and tensions, at any given time*.

MEXICA allows mixing of the two types of representation, namely actions and contexts. So, it is possible to associate the story context at time = n with the action at time = $n + 1$. That is, we can link the status of the narrative world at any given time with a deed to be performed. For example, if we associate the story context 1 and the action 2, we can represent the following information: when the health of a character is at risk (in this case the knight), *it makes sense* that a second character (in this case the princess) arrives and cures him. If we associate the story context 2 and the action 3, we can represent the following information: when character A is very grateful to character B, *it makes sense* that a third character C kidnaps character B. Thus, these structures represent CSK that describes what to do in a given context.

MEXICA creates its knowledge base using a similar procedure. The user supplies, together with the dictionary of story actions, a set of well-formed narratives known as the previous stories. The system takes the first previous story and generates its story contexts. Then, MEXICA copies each story context into what is known as a *contextual structure*, substitutes characters for variables and then associates with it the next action in the tale. Figure 12.4 shows two contextual structures that arise from the story contexts at time = 1 and time = 5 shown in Fig. 12.3. The first structure indicates that when character A is wounded, it is logical that some other character arrives and cures A. The second structure records the following information: when character A is very grateful to character B, and character A and B hate character C, and character C has character B as a prisoner, it makes sense that character A rescues character B. MEXICA repeats the same procedure for all previous stories. A *contextual memory* is defined as the group of contextual structures that an agent has.

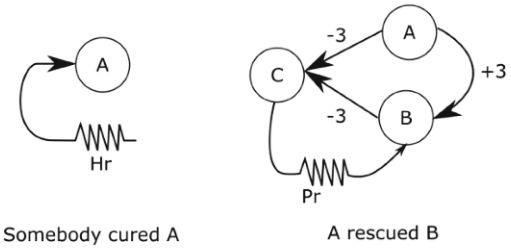


Fig. 12.4 Contextual structures employed by MEXICA.

If the system is fed with enough tales, each contextual structure might have associated with it several possible next actions to be performed. For instance, in the story depicted in Fig. 12.3, one character has an accident and then another character cures him; now, imagine a second previous story, where one character has an accident and then other characters takes the opportunity to mug him, and a third previous

story, where one character has an accident and then a second character asks for help. Rather than creating three different contextual structures, the system builds only one and gathers together the three possible actions. Figure 12.5 shows the result after processing our three imaginary previous stories. This structure indicates that, when a character is wounded, a logical action to progress the narrative is that a new character arrives and either cures him, mugs him or screams for help.

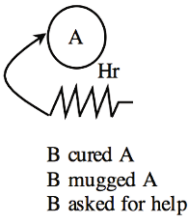


Fig. 12.5 A contextual structure with three possible next actions to be performed.

Thus, MEXICA's knowledge base is built from the content of the previous stories. If we provide the system with a group of tales that reflect social traditions, some aspects of such traditions might be represented in the contextual memory.

After creating its knowledge base, MEXICA generates new plots through engagement and reflection cycles. It is out of the scope of this chapter to provide details of this process (readers interested in the specifics of the model are recommended to consult the references). The following list offers a general view of such a process:

i The user provides an initial action.
ii The system triggers the postconditions of the current action, updates the story world context and uses it as cue to probe its memory.
iii MEXICA attempts to match a contextual structure that is equal or similar to the story context. If the context is equal to at least 50% of the contextual structure, the system considers that they are similar.
iv The system retrieves all the actions associated with the matched structure and selects one at random as the next deed in the tale. It goes back to step ii until three actions are generated; then, it proceeds to step v.
v The system switches to reflection to evaluate the interestingness, novelty and coherence of the story in progress and, if necessary, to modify it. Then it goes back to engagement (step ii) and the cycle continues until the story is finished.

Next, I present an example that illustrates how a story is generated.

Imagine that we have a contextual memory like the one shown in Figure 12.6. The structure in Fig. 12.6a represents that when character A fancies character B it makes sense that A falls in love with B; Fig. 12.6b represents that when character A is in love with B, but B dislikes A, it makes sense that B ignores A; Fig. 12.6c represents that when character A dislikes B, it makes sense that A makes fun of B. Remember that A and B are variables that can be substituted by any character in the tale.

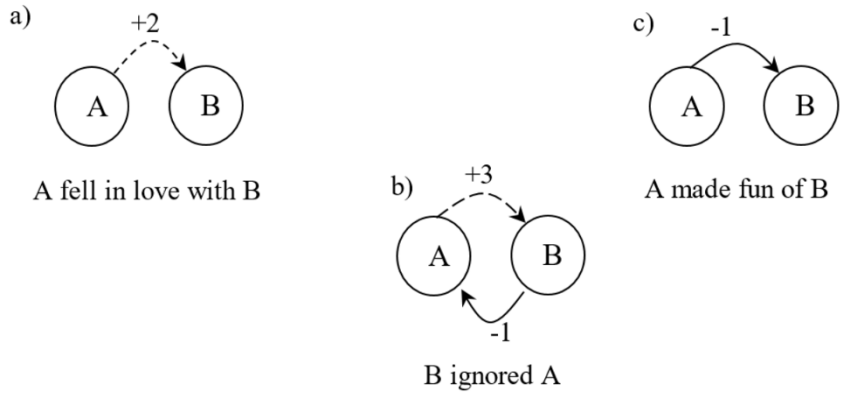


Fig. 12.6 An example of a contextual memory.

Now, let us start to develop a new tale with the following initial action: 'Jaguar Knight felt very attracted to the princess'. This action produces the story context shown in Figure 12.7 at time = 1. MEXICA uses this story context as a cue to probe its memory, and it is able to match a contextual structure that is identical to it (see Fig. 12.6a). Thus, character A is substituted by Jaguar Knight and character B by the princess, the system selects as the next action in the story in progress 'Jaguar Knight fell in love with the princess' and the story context is updated (see Fig. 12.7 at time =2).

Next, MEXICA employs the current story context (at time = 2) as a cue to probe its memory, but this time it cannot find a structure that is equal to the context. So, it matches the structure depicted in Figu. 12.6b, which is 50% alike. Thus, character A is substituted by Jaguar Knight, character B by the princess, the system selects as the next action in the story in progress 'Princess ignored the jaguar knight' and the story context is updated as shown in Fig. 12.7 at time = 3 (although the knight is still in love with the princess, he dislikes her because she is ignoring him; that is why the knight develops an emotional link of type 1 and intensity −1 towards the princess).

Again, the system probes its memory and matches the similar structure shown in Fig. 12.6c, which is 50% alike. Thus, character A is substituted by Jaguar Knight and character B by the princess, the system selects as the next action in the story in progress 'Jaguar knight made fun of the princess' and the story context is updated as shown in Fig. 12.7 at time = 4 (because the knight is making fun of the princess she now also dislikes him; that is why the princess develops an emotional link of type 1 and intensity −1 towards the knight).

At this point the system switches to reflection to evaluate the story in progress. Then, it goes back to engagement and the cycle continues.

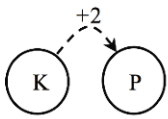
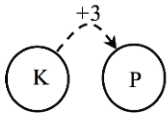
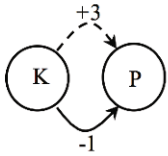
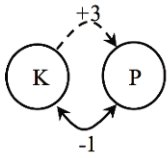
Time	Action	Story-Context
1	Jaguar knight felt very attracted to the princess.	
2	Jaguar knight fell in love with the princess	
3	Princess ignored the jaguar knight	
4	Jaguar knight made fun of the princess	

Fig. 12.7 An example of a story in progress (K stands for Jaguar Knight and P for the princess).

12.5 Discussion

The production of coherent sequences of actions is one of the most challenging goals for models of plot generation. Some researchers have employed predefined story structures to sort this problem out. That requires that the designer (or the user) of the system specifies in advance core features of the tale that the system will develop. As a result, the outputs of the automated storyteller may be rigid and predictable. But most importantly, this approach limits our contribution to the comprehension of a crucial characteristic of creative human beings. I believe that the way MEXICA represents CSK might be an alternative for producing chains of events that make sense.

12.5.1 Representation of Knowledge

MEXICA's knowledge base is built from the dictionary of story actions and the previous stories provided by the user. Both are text files that can be easily updated.

The dictionary allows one to define and modify social and logical preconditions and postconditions. Thus, using the same system, it is possible to represent a community where characters react violently to situations (postconditions with emotional links of intensity -3) or a different one where characters react calmly to the same situations (postconditions with emotional links of intensity 0 or -1), and so on. Also, it is possible to include logical CSK; for example, two characters must be in the same position to interact or a character must be ill to be cured.

The previous stories are employed to construct contextual structures that are associated with possible behaviours that make sense; they represent the agent's experience. Such structures reflect some of the social and cultural traditions embedded in the previous stories. For instance, if the tales include situations where colleagues are very close and support each other, the system produces contextual structures that comprise relations depicting friendship (emotional links of type 2) and are associated with actions that promote cooperation.

Because the knowledge base is built from the dictionary of story actions and the previous stories, it is simple to update an agent's experience. Thus, it is easy to run different tests.

Evidently, human social behaviour is much more complex than the representation employed in MEXICA. For instance, the system is not capable of dealing with situations where an action that makes sense in a given context might be inexplicable when the context changes, or situations where what is considered as specialised knowledge by one group of individuals might be considered as common-sense knowledge by a different group. Although we are working in that direction (se e.g. (Guerrero & Pérez y Pérez, 2014; Pérez y Pérez, 2015a; Pérez y Pérez, Castellanos, Ávila, Peñalosa, & Negrete, 2011)), there is much work left to do. Nevertheless, this work illustrates how CSK associated with communal traditions can be represented in a plot generator.

12.5.2 Employing CSK to Generate Coherent Sequences of Actions

In general, MEXICA employs actions with a reduced level of description. The use of emotional links and tensions between characters provides a very flexible and powerful tool for representing CSK for plot generation. Thus, its preconditions and postconditions can be classified as having a tendency to be open and as being intangible. That is why contextual structures can be associated with diverse possible next actions to be performed (see Fig. 12.5). In this way, the same context can drive the tale in completely different directions. Nevertheless, it is necessary to improve MEXICA to represent concrete preconditions and postconditions in order to compose more appealing tales.

The story context, and therefore the contextual structures, *encapsulates in a flexible representation the core events that have happened in the narrative* so far (see Figs. 12.3 and 12.7); this is one of the main contributions of the system. This is an example of how to use social and cultural CSK to generate sequences of actions that

make sense, avoiding the use of predefined story structures. The use of engagement and reflection illustrates how to exploit this type of knowledge. During engagement, when MEXICA is able to match a pattern that is equal to the context, the system produces a consistent chain of events (see Figs. 12.6a and 12.7 at time = 1). However, if the program encounters only structures that are identical, it starts to reproduce the previous stories. To avoid this condition, the system also matches schemes that are similar to the context (see Figs. 12.6b, 12.6c, and 12.7 at time = 2 and Fig. 12.7 at time = 3). In this way, the narrative in progress maintains a degree of coherence and at the same time produces novel outputs. If necessary, during reflection the system modifies the material generated so far to ensure that it is consistent.

12.6 Conclusions

> It is doubtful that the facts of the common sense world can be represented adequately by production rules... Much present AI research concerns how to represent facts in ways that permit them to be used for a wide variety of purposes. (McCarthy, 1984)

At the beginning of this chapter I claimed that computational creativity must contribute to the understanding of the creative process. The ideas described in this text provide a framework that allows one to analyse important features of computer models of storytelling; such a proposal facilitates the understanding of this phenomenon and provides a perspective that might influence future developments. In the chapter, I have described how story actions have different levels of description associated with them; in the same way, I have classified preconditions and postconditions as social or logical, and explained two of their attributes: granularity and level of abstraction. All of these characteristics are related; if we reflect on how they interact and what their possible effects on automated plot generation are, and we contrast these ideas with other approaches, we will be able to develop better systems. In particular, I have emphasised the relevance of representing social CSK because what else are narratives but an expression of our deepest need to communicate with each other? Thus, any computer model of plot generation demands adequate representations of social knowledge.

MEXICA illustrates how these ideas can be implemented; but I am certain that there are multiple possibilities for exploiting such concepts (see, for example, how we use the same model in our work on computer models of design in (Pérez y Pérez, Aguilar, & Negrete, 2010) and (Pérez y Pérez, González de Cossío, & Guerrero, 2013)). MEXICA uses emotional links and tensions between characters to represent CSK that permits it to generate sequences of actions that make sense. The analysis presented in this chapter suggests this is a promising approach. However, there is much work left to do. Some of the improvements that the system requires have already been mentioned. Furthermore, it is necessary to keep on investigating powerful ways to represent CSK, to design agents able to exploit the whole range of granularity and levels of abstraction during narrative generation, to develop collaborative models that allow storytellers to learn from interactions with other agents (such as with the system

described in this volume by Saunders (2019)); and so on. Also, it is necessary to incorporate into the discussion topics like intentionality and autonomy (see (Ventura, 2019) in this volume).

This chapter has focused on describing the use of CSK to generate coherent sequences of actions. However, that is not enough for plot generation. MEXICA also includes strategies to produce interesting and novel tales. The system has been evaluated by asking subjects to assess the characteristics of the stories produced by the agent. We encourage readers to study the references provided.

Narrative generation is essential for the development of cognitive and social skills. Computational creativity offers a great opportunity to contribute to its understanding. I hope this chapter inspires students and researchers to pursue this fascinating area of study.

Acknowledgements This research was sponsored by the National Council of Science and Technology of México (CONACYT), project number 181561.

References

Anacleto, J., Lieberman, H., Tsutsumi, M., Neris, V., Carvalho, A., Espinosa, J., … Zem-Mascarenhas, S. (2006). Can common sense uncover cultural differences in computer applications? In M. Bramer (Ed.), *IFIP International Conference on Artificial Intelligence in Theory and Practice* (Vol. 217, pp. 1–10). Artificial Intelligence in Theory and Practice. Boston: Springer.

Bremond, C. (1996). La lógica de los posibles narrativos (trad.) In *Análisis estructural del relato* (pp. 99–121). Originally published as La logique des possibles narratifs, Communications, 1966, No. 8. Mexico, D.F: Ediciones Coyoacan.

Eubanks, P. (2004). Poetics and narrativity: How texts tell stories. In C. Bazerman & P. Prior (Eds.), *What writing does and how it does it: An introduction to analyzing texts and textual practices* (pp. 33–56). Mahwah, New Jersey: LEA.

Gervás, P. (2009). Computational approaches to storytelling and creativity. *AI Magazine*, *30*(49–62).

Gervás, P. (2019). Exploring quantitative evaluations of the creativity of automatic poets. In T. Veale & F. A. Cardoso (Eds.), *Computational creativity: The philosophy and engineering of autonomously creative systems* (pp. 273–302). Springer.

Guerrero, I., & Pérez y Pérez, R. (2014). Social MEXICA: A computer model for social norms in narratives. In *Proceedings of the 5th International Conference on Computational Creativity, ICCC–2014* (pp. 192–200). Ljubljana, Slovenia.

Jordanous, A. (2019). Evaluating evaluation: Assessing progress and practices in computational creativity research. In T. Veale & F. A. Cardoso (Eds.), *Computational creativity: The philosophy and engineering of autonomously creative systems* (pp. 209–234). Springer.

Lieberman, H. (2008). Usable artificial intelligence requires common sense knowledge. In *Workshops and courses: Usable artificial intelligence*, Held in conjunction with CHI 2008, Florence, Italy.

McCarthy, J. (1984). Some expert system need common sense. In H. Pagels (Ed.), *Computer Culture: The Scientific, Intellectual and Social Impact of the Computer* (Vol. 426, pp. 129–137). New York Academy of Sciences.

McCarthy, J. (1989). Artificial intelligence, logic, and formalizing common sense. In R. Thomason (Ed.), *Philosophical Logic and Artificial Intelligence* (pp. 161–190). Dordrecht, Kluwer.

Montfort, N., & Pérez y Pérez, R. (2008). Integrating a plot generator and an automatic narrator to create and tell stories. In *Proceedings of the 5th International Joint Workshop in Computational Creativity, IJWCC 2008, Universidad Complutense de Madrid, Spain* (pp. 61–70).

Pérez y Pérez, R. (1999). MEXICA: A Computer Model of Creativity in Writing. DPhil dissertation, University of Sussex, U. K.

Pérez y Pérez, R. (2007). Employing emotions to drive plot generation in a computer-based storyteller. *Cognitive Systems Research*, *8*(2), 89–109. doi:10.1016/j.cogsys.2006.10.001

Pérez y Pérez, R. (2015a). A computer-based model for collaborative narrative generation. *Cognitive Systems Research*, (36-37), 30–48.

Pérez y Pérez, R. (2015b). From MEXICA to MEXICA-impro: The evolution of a computer model for plot generation. In T. Besold, M. Schorlemmer, & A. Smaill (Eds.), *Computational creativity research: Towards creative machines* (7). doi:10.2991/978-94-6239-085-0_13

Pérez y Pérez, R. (2015c). MEXICA-impro: Generación automática de narrativas colectivas. In R. Pérez y Pérez (Ed.), *Creatividad computacional* (pp. 95–110). UAM-Cuajimalpa-Patria.

Pérez y Pérez, R., Aguilar, A., & Negrete, S. (2010). The ERI-Designer: A computer model for the arrangement of furniture. *Minds and Machines*, *20*(4), 533–564.

Pérez y Pérez, R., Castellanos, V., Ávila, R., Peñalosa, E., & Negrete, S. (2011). MEXICA-impro: Ideas para desarrollar un modelo computacional de improvisación. *CIENCIA ergo sum*, *18*(1), 35–42.

Pérez y Pérez, R., González de Cossío, M., & Guerrero, I. (2013). A computer model for the generation of visual compositions. In *Proceedings of the 4th International Conference on Computational Creativity, ICCC-2013, Sydney, Australia* (pp. 105–112).

Pérez y Pérez, R., & Sharples, M. (2001). MEXICA: A Computer Model of a Cognitive Account of Creative Writing. *Journal of Experimental & Theoretical Artificial Intelligence*, *13*(2), 119–139.

Pérez y Pérez, R., & Sharples, M. (2004). Three computer-based models of storytelling: BRUTUS, MINSTREL and MEXICA. *Knowledge-Based Systems*, *17*(1), 15–29.

Ritchie, G. (2019). The evaluation of creative systems. In T. Veale & F. A. Cardoso (Eds.), *Computational creativity: The philosophy and engineering of autonomously creative systems* (pp. 157–192). Springer.

Saunders, R. (2019). Multi-agent based models of social creativity. In T. Veale & F. A. Cardoso (Eds.), *Computational creativity: The philosophy and engineering of autonomously creative systems* (pp. 303–325). Springer.

Sharples, M. (1999). *How We Write? Writing as Creative Design*. London: Routledge.

Ventura, D. (2019). Autonomous intentionality in computationally creative systems. In T. Veale & F. A. Cardoso (Eds.), *Computational creativity: The philosophy and engineering of autonomously creative systems* (pp. 49–69). Springer.

Chapter 13
Exploring Quantitative Evaluations of the Creativity of Automatic Poets

Pablo Gervás

Abstract The purpose of this chapter is twofold: to show the practical applications of theoretical evaluation measures designed to capture the degree of creativity of a program, and to use the results to evaluate an effort to develop an automatic Spanish poet. Existing efforts in the development of automatic poets are described, and the implications of their particular architectures for the evaluation issues discussed are considered.

13.1 Introduction

The community of researchers devoted to the study of creativity has, over the years, grown from a small group of people who worked on isolated projects to reach an important number of groups addressing issues in different domains and with different areas of focus. In this process, there is a need for some kind of quantitative means of evaluating the quality or the efficiency of a creative system. The need to have objective measures is crucial in a general sense if we are to achieve the development of testable and comparable solutions to the problems that are being faced. In a field with as much subjective content as that of creativity, it becomes paramount not only to define some means of establishing quantitative measurements, but also to apply such measurements systematically to the solutions designed at each stage, in order to obtain from them guidance and stable references on which to base further development.

In recent times, research efforts in the field of creativity have produced a number of systems that attempt tasks that were previously considered to be too creative for computers to tackle, such as musical composition, theory formation, and poetry writing. The resulting increase in interest from the research community has produced

Pablo Gervás
Instituto de Tecnología del Conocimiento, Universidad Complutense de Madrid, Spain.
e-mail: pgervas@ucm.es

T. Veale, F. A. Cardoso (eds.), *Computational Creativity*, Computational Synthesis and Creative Systems, https://doi.org/10.1007/978-3-319-43610-4_13

the very beginnings of a theoretical body of work on the evaluation of creativity.[1] If the field is to progress steadily, the next step ahead should be to apply these initial theoretical efforts to the practical systems being developed. The purpose of this endeavour should be twofold. On one hand, it should provide quantitative metrics for the creative behaviour of systems that may play a role in guiding subsequent design and development efforts. On the other hand, it should at the same time constitute a test of the suitability of the theoretical evaluation methods that have been proposed so far. While both theoretical and practical advances are valuable in a field as young as the study of creativity as it relates to computer programs, it is really in bringing together theory and practice that positive progress will be consolidated in the field. Practical systems should be tested according to the theories that are being put forward, and theoretical proposals should be applied to real cases to see if they adequately address the issues that are of import in the development of systems.

The present chapter brings together a number of theoretical proposals put forward in the past few years and the practical creative systems developed in the past for the particular domain of poetry composition. Important issues concerning this particular domain are discussed, the different approaches are compared and different proposals for the evaluation of creative behaviour are tried out against the results from one particular system.

13.2 Existing Formalisations of Creativity Measurement

If creativity and engineering are to collaborate successfully, some explicit way of measuring the activity involved in creativity must be established. Boden (1990) distinguished between *H-creativity* – the result is absolutely new in historical terms – and *P-creativity* – the result is new for the creator, independently of whether it has been done before. In the face of this type of distinction, it is important to establish criteria that can be applied to a system and which take into account not only what is being created but also what was already available to the program when it started operating. Additionally, measurements must take into account the need for a balance between creating artefacts that meet the general requirements within a given domain (are typical of the genre) and creating artefacts that are innovative (bring something new to the domain).

There are currently several proposals about how creativity might be measured quantitatively, at least indirectly if not directly, and in terms of certain qualifying functions for the specific domain in which the creative system to be evaluated is operating.

[1] A review of the evolution of this body of work since the original version of this chapter was written may be found by the interested reader in the chapters by Anna Jordanous (2019) and Graeme Ritchie (2019) in the present book.

13.2.1 Assessing Creativity Based on How Good and How Typical the Results Are

Ritchie (2001) provides an initial set of relevant concepts and 14 criteria based on those concepts for deciding whether a program is creative or not, which constitute a strong starting point for discussion. Two basic aspects are relevant: *novelty* (to what extent is the item produced dissimilar to existing examples of that genre) and *quality* (to what extent is the item produced a high-quality example of that genre). Ritchie opts to consider novelty in terms of a complementary property, the typicality of a given artefact, in terms of how similar it is to existing ones. To measure these aspects, two functions are introduced: *typ*, which rates the typicality of a given item (to what extent the item is typical), and *val*, which rates its quality (to what extent the item is good). These functions take the form of rating schemes, which assign points in the interval [0,1] for a given property.

Another important issue that affects the assessment of the creativity of creative programs is the concept of the *inspiring set*, the set of (usually highly valued) artefacts that the programmer is guided by when designing a creative program. According to this vision, the construction of a creative program follows a sequence of steps that can be instantiated for any particular case:

1. Select a set of basic items which are to guide the construction of the creative program (the *inspiring set*).
2. Map from the inspiring set to a program.
3. Establish a generating procedure (which, given a tuple of initial data values produces a set of basic items) and possible parameters for this procedure (a tuple of sets, each set being the range of one parameter of the generating procedure).

Program construction therefore follows a basic scheme divided into two processes: one of selection (from the basic items get the inspiring set) and one of construction proper (define the initial data ranges and procedure, initialise, run, and obtain a result set). The initialisation consists of selecting a choice of initial parameters, based on the inspiring set and the rating schemes used. A *run of the program* is understood as a set of initial parameters together with the set of results.

Fourteen criteria are provided, relating the inspiring set and the set of results (and the corresponding subsets defined by applying to it the valuation functions mentioned above). These criteria are intended as a box of tools from which to pick and choose a selection for a particular purpose. A brief overall description of their intention is given in Table 13.1. Specific parameters are provided in the mathematical formulation of these criteria to control what is actually meant by "a reasonable proportion".

Ritchie (2007) revised this initial set of criteria in view of some issues that had arisen as a result of applying them in practice. In this revision, criteria 8 and 10 were revised to solve some problems arising when none of the artefacts in the inspiring set were replicated in the results. A number of additional criteria were added to cover cases related to proportions involving highly valued or highly typical items in the

part of the results that were novel. The revised set of criteria is described in Table 13.2.[2]

Table 13.1 Description of basic criteria

Criterion	Description
1	All elements in the result should be reasonably typical
2	A reasonable proportion of the results should be very typical
3	All elements in the result should be reasonably good
4	A reasonable proportion of the results should be very good
5	A reasonable proportion of the very typical results should be very good
6	A reasonable proportion of the results should be very good and not very typical
7	A reasonable proportion of the not very typical results should be very good
8	There should be a reasonable ratio between the very good and not very typical results and the very good and very typical results
9	A reasonable proportion of the inspiring set should appear in the results
10	A reasonable proportion of the results should not appear in the inspiring set
11	Results not in the inspiring set should be typical
12	Results not in the inspiring set should be very good
13	A reasonable proportion of the results should not appear in the inspiring set and should be very typical
14	A reasonable proportion of the results should not appear in the inspiring set and should be very good

Table 13.2 Description of revised criteria

Criterion	Description
1	All elements in the result should be reasonably typical
2	A reasonable proportion of the results should be very typical
3	All elements in the result should be reasonably good
4	A reasonable proportion of the results should be very good
5	A reasonable proportion of the very typical results should be very good
6	A reasonable proportion of the results should be very good and not very typical
7	A reasonable proportion of the not very typical results should be very good
8a	There should be a reasonable ratio between the very good and not very typical results and the very good results
9	A reasonable proportion of the inspiring set should appear in the results
10a	A reasonable proportion of the results should not appear in the inspiring set
11	Results not in the inspiring set should be typical
12	Results not in the inspiring set should be very good
13	A reasonable proportion of the results should not appear in the inspiring set and should be very typical
14	A reasonable proportion of the results should not appear in the inspiring set and should be very good
15	Very typical results should be a high proportion of the novel items
16	Very good results should be a high proportion of the novel items
17	Good and typical should be a high proportion of the novel items
18	Good and untypical results should be a high proportion of the novel items

[2] The mathematical formulation of criterion 10 was changed to avoid division by zero when no results appeared in the inspiring set, but its paraphrase in the current table is still applicable.

13.2.2 Evaluating the Degree of Fine Tuning

Colton, Pease, and Ritchie (2001) described the effect on the perceived creativity of a program of the amount of knowledge that is taken as a starting point in whatever generation process it carries out. This was done based on the work of Ritchie (2001), by refining the criteria proposed there in terms of measurements designed to capture the degree to which a particular creative program involves *fine tuning*, in the sense of tailoring the design of the program to produce a particular kind of output.

To achieve this, the following concepts are introduced:

- The *Output set* O_K: the set of items produced by a program using knowledge K.
- The *Reinventions set* R_K: the set of items already present in the inspiring set reproduced by a program using knowledge K.
- The *Creative set* C_K: the set of valuable items produced by a program using knowledge K, excluding those in the inspiring set.
- The *Dependency set* $D_{K'}$ of a subset K' of the input knowledge K: that part of the valuable results which will be missing from the output if K' is removed from K.

For a particularly creatively useful K' – in the sense that removing it from the input knowledge reduces the valuable results – we can say that K' is fine tuned if

$$|D_{K'} \cap R_K| > 0$$

and

$$|D_{K'} \cap C_K| = 0$$

This corresponds to cases where the contribution of K' to high-value output is restricted to replicating elements that were already present in the inspiring set. When there are at least some high-valued items contributed by K' ($|D_{K'} \cap C_K| > 0$) the following definition gives the measure of how fine-tuned K' is:

$$ft(K') = \frac{|D_{K'} \cap R_K|}{|D_{K'} \cap C_K|}$$

This returns a value greater than 1 if K mainly rediscovers already-known artefacts rather than finding new ones of value, and it returns values of 1 or less otherwise.

13.3 Automatic Generation of Poetry: Approaches and Evaluation Issues

The automatic generation of poetry attracted a certain amount of interest in the recent past. The complexity of the task, involving several levels of language use (phonetics, lexical choice, syntax, semantics, discourse structuring, ...) gives rise to a domain of artefacts of high complexity, where a considerable amount of input

knowledge is required. The various approaches that have been attempted so far differ considerably in the amount of input knowledge that the creative programs are provided with to carry out their task. In all cases, the input knowledge provided to the poetry generation system also constrains the type of output that can be produced. The input knowledge can range over particular fragments of text to be reused – phrases, lines, sentences or templates consisting of ready-written text with gaps left to be filled during construction – particular syntactic structures understood as sequences of part-of-speech (POS) tags, specific semantic relations to be observed between the words used in the poem, specification of a desired emotional effect, specific messages encoded as semantic descriptions, or conformance to known sequencing of words as expressed by *n*-gram models of language. These various features will be used to analyse the systems reviewed below, because they jointly affect which aspects of the text need to be represented in each case. By following this criterion for establishing the order of presentation of existing systems, we have to accept that their relative chronology is not respected.

The different approaches to input for the automatic generation of poetry that are considered below each give rise to different issues that may have to be taken into account when designing suitable evaluation methods. Because in most cases the input for poetry generation systems is drawn from a set of examples of the desired target artefacts, the choice of format for the input has a very high impact on the relation between the output of any given system and its inspiring set. According to Ritchie's definition of the inspiring set, the set of examples of the desired target artefacts constitute clear members of the inspiring set. The classification of systems proposed here therefore constitutes a good framework in which to consider some of the issues that need to be discussed in this chapter. For this reason, each of the subsections below dedicated to a particular approach to poetry generation includes a small discusion of how the proposed evaluation methods might be applicable to results obtained by means of that approach.

13.3.1 Starting from Text-Based Templates

A large number of poetry generation methods rely on reusing fragments of text, usually compiled from a corpus of samples of poetry. These methods apply procedures for dividing the reference samples into smaller units of text that are then reused, either recombined among themselves or combined with similar fragments of text obtained from alternative sources.

13.3.1.1 Poetry Generation Based on Textual Templates

Early approaches to systematising the construction of poetry occurred in the literary world independently of the drive for computational creativity. The work of Queneau (1961) constitutes a combinatorial approach to poem generation where a large number

of poems are obtained simply by interchanging the lines of a set of poems, always respecting the same relative position of each line within a set template stanza. The lines in each particular position were all carefully crafted so as to be compatible with all other possible continuations.

The ALAMO group (*Atelier de Littérature Assistée par la Mathématique et les Ordinateurs*) undertook the task of applying modern computer power to operationalise the efforts of OULIPO (*Ouvroir de Littérature Potential*) (Oulipo, 1988), a previously existing group which had explored the algorithmic construction of literature. The ALAMO group has been generating poems in French automatically for some time. Amongst other activities dedicated to the promotion of the use of computers for literary creativity, this group have presented a number of text generation programs (*literaciels*) which they use for animating literary workshops.

A number of examples of these programs are described in their website.[3] From the information provided (example of results, brief description of the methods followed) the general idea behind these programs seems to be to identify a set of texts, words and valid transformations that allow automatic generation of a fixed number of alternative versions of a given poem. For instance, they describe the construction method for *Rimbaudelaires*, sonnets obtained by combining the sentence structure of sonnets by Rimbaud with the vocabulary of the poetry of Baudelaire. A poem shell is obtained by cutting out the nouns, verbs and adjectives from a given sonnet by Rimbaud. Metrically matching words from the vocabulary of Baudelaire are then used to fill the resulting gaps, following "strong syntactic and rhythmic constraints". Although their methods seem to be based on identifying basic poem structures which allow a number of variations (in terms of substitution of words for size-matching equivalents at given substitution hotspots), the resulting effect is striking, and does give the impression of a reasonably articulate poet at work.

The corpus-based approach of Toivanen, Toivonen, Valitutti, and Gross (2012) also reuses text, but it introduces small changes in terms of word substitution. It relies on two different corpora, one to establish constraints on form and one to establish constraints on content. To set constraints on content, a word association network is built from a corpus consisting of the Finnish pages of Wikipedia. To set constraints on form, fragments are selected from a corpus of old Finnish poetry. A word substitution method is applied: start from a given fragment from the poetic corpus, and replace selected words with syntactically compatible words obtained from the word association network.

This approach was refined further by Toivanen, Gross, and Toivonen (2014) where additional constraints are imposed on the words being used to replace the original words: candidate fillers must come from foreground associations obtained for a given input news item, i.e. words in the current news item not already associated with a prior news item on the same topic. When associations are employed to select the desired words, an additional level of semantic-related criteria is used as a filter for the retrieval of words.

[3] http://www.alamo.free.fr/

The FloWr system (Charnley, Colton, & Llano, 2014), intended for implementing creative systems as scripts over processes and manipulated visually as flowcharts, has been used to build poetry generation systems by combining a number of strategies. *Word selection strategies* are used to find words within a selected range of frequency of use, trim the set down to a particular syntactic category, and restrict the selection further to words with a particular sentiment. *Retrieval strategies* are employed to retrieve tweets containing a word picked randomly from the filtered set. *Composition strategies* are applied to collate the resulting fragments into stanza-like groupings. Ultimately, the FloWr system relies on reuse of complete tweets or sentences extracted from tweets, with results emerging on the basis of the various procedures employed for selection and filtering of the input material. This system relies on a two-tier approach to poem building: one process for building candidate lines, and one for collating selected candidates into stanzas.

A similar solution is applied in the Pemuisi system (Rashel & Manurung, 2014) where templates for lines are obtained by abstracting content words from a corpus of Indonesian poems, and then filled with keywords extracted from newspapers with an intermediate semantic expansion process applied to improve variation. Selection of slot fillers is treated in this case as a constraint satisfaction problem; constraint satisfaction (at the poem level) is also used to select a set of lines to e combined into a poem. Pemuisi also follows the idea of articulation in two tiers: the task of building a poem is divided into a procedure for constructing lines and a procedure for combining valid lines into poems.

13.3.1.2 Assessing Outputs Obtained from Text-Based Templates

The method of production described for *Rimbaudelaires* differs from the other approaches described in that it is template based, in the sense that there is a husk of the poem which contains a number of words already in fixed positions, and which cannot be modified by the generation process. The degree of freedom involved is very low. In contrast, other systems allow modification at word level in all positions in the poem.

The poetry produced by the ALAMO group should rate highly on typicality and quality, but low on originality. This is a particularly good case for applying criteria related to a comparative study of how much of the result set is included in the inspiring set. It is possible that all of the result set for systems based on this architecture may be present in the inspiring set from the start.

Similar considerations should apply to all systems that operate on text-based templates. In each case, the values for Ritchie's criteria if applied to this kind of system are likely to be severely affected by parameters such as the size of the elementary unit chosen for templating (a line, a sentence, a stanza or the whole poem), the relative size of the fixed part of the template compared with the gaps that can be filled in with new words. In systems with multitiered construction procedures, the effect of the regularities introduced by the template may be considerably reduced.

Each tier allows a further degree of recombination of the original material, leading to less similarity of the results to the inspiring set.

13.3.2 Starting from POS Tag Sequences

A different approach involves stripping all the words from the reference fragment for form, and retaining only its sequence of POS tags, and then filling those in with words from a different source, with criteria for appropriateness to drive the process.

13.3.2.1 Poetry Generation Based on Syntactic Templates

An example of this approach is the early version of the WASP system (Gervás, 2000b), which draws on prior poems and a selection of vocabulary provided by the user to generate a metrically driven recombination of the given vocabulary according to the line patterns. A set of plausible sequences of POS tags for lines – extracted from the set of prior poems – together with the selection heuristics based on metrics, constrain the form, and the vocabulary provided by the user constrains the content at a lexical level.

This is also the approach followed by Agirrezabal, Arrieta, Hulden, and Astigarraga (2013), which relies on extracting POS tag sequences for lines in a given corpus of poems – using different corpora for different lengths of line in the stanza – finding the most commonly used such POS tag patterns for lines, and filling those chosen patterns with new ones, based on criteria at two levels, one syntactic – matching POS tag and morphology – and one semantic. Several semantic criteria were considered but best performance was obtained by replacing only nouns with other semantically related nouns, ensuring that morphological information from the original word was transferred to the substitute.

A similar procedure but with more refined constraints on form was used by Toivanen, Järvisalo, and Toivonen (2013). Instead of taking an actual fragment and replacing some of the words in it with new ones corresponding to the desired content, the selected fragment is stripped down to a skeleton consisting only of the POS tags of each line, and words corresponding to the desired content are used to fill this skeleton in, while obeying a complex set of constraints. Constraints can be established based on rhyme, number of syllables per line, occurrence of particular words or even syntax, and they can be loosened so that rather than being binary they allow for grading of the solutions in terms of how well they satisfy the constraints. The solution is then sought by applying Answer Set Programming to search for optimal assignments of candidate words that optimise this grading.

13.3.2.2 Assessing Outputs Obtained from POS Tag Sequences

The initial WASP system presented by Gervás (2000b) constructed poems by taking lines in prior poems as the basic building units for adaptation. This procedure resulted in a very agile construction mechanism, tailored to ensure strict metrical correctness and focusing closely on rhyme, but led to poor results from a syntactic and semantic point of view. Regarding evaluation, such systems would require a definition of the input knowledge in terms of the set of line patterns being considered, and the additional vocabulary. The inspiring set in each case was explicitly defined as the set of original poems from which the line patterns are drawn. It would still have to be decided whether the introduction of additional vocabulary outside the words appearing in those poems should be considered as an extension of the inspiring set. Maybe the concept of an inspiring set should be redefined to include this sort of situation.

13.3.3 Starting from Prose-to-Verse Matched Pairs

A number of systems have addressed the problem of generating verse as a problem of how to convert a given message expressed in prose into a poem that conveys a similar meaning. To achieve this, they rely on a case base of pairs of text fragments, where the first element of the pair expresses the desired message in prose and the second element of the pair expresses the message in verse.

13.3.3.1 Poetry Generation as Prose-to-Verse Conversion

An evolution of the WASP system (Gervás, 2000a, 2001a, 2001b) used case-based reasoning (CBR) (Aamodt & Plaza, 1994) to build verses from an input sentence by relying on a case base of matched pairs of prose and verse versions of the same sentence. Each case was a set of lines of verse associated with a prose paraphrase of their content. An input sentence was used to query the case base, and the structure of the verses of the best-matching result was adapted into a verse rendition of the input. The syntactic structure of the solution – the sequence of POS tags for the verse version, including the corresponding line breaks – was used as template to be filled in with words from the user query, guided by metric restrictions and falling back on the actual words of the poem in case of a mismatch. As the solutions did not necessarily match full stanzas or even complete lines, an additional tier of construction was included to combine the verse renditions of each input sentence into complete poems.

From the CBR point of view, the main difference between the various systems is that the ASPID system (Gervás, 2000a) operates on line-sized cases, which it composes to build coherent stanzas, and the ASPERA system (Gervás, 2001a, 2001b) operates on stanza-sized cases. Additionally, ASPERA is a more complex system, having extra modules for user interaction. These modules request additional input

parameters from the user concerning the setting, mood and length of the intended message, and apply a knowledge-based system to filter an appropriate starting set of cases and vocabulary.

In general terms, the construction modules of both systems start from an initial target content provided by the user (the intended message), and retrieve a case to later use the solution as a seed structure to fill in. The intended message is used as the main source for the words required to fill in the case, and the case itself is used as the default source. An additional vocabulary (also provided by the user) is used as an intermediate source.

13.3.3.2 Assessing Outputs Obtained from Prose-to-Verse Matched Pairs

Regarding the kind of evaluation discussed here, there is one particular issue that needs to be taken into account. The intended message provided by the user, by allowing the user some control over the ingredients that will be used to produce the final result, may affect the question of whether the system is fine-tuned. By exercising this control to guide the system towards particular results, the system can be forced to produce results that are very close to the inspiring set.

In the case of ASPERA, further control features are provided in the form of initial basic data that act as system parameters. These control the way in which the system splits the target content among the lines in the chosen stanza, the kind of similarity employed by the system to retrieve cases to be used as a seed for the construction process, the number of syllables required per line, the number of lines in the stanza, and the amount of variation in the relative position of a word between the target content and the final result. They are used to provide the system with a certain degree of freedom which can be controlled by the user. This feature results in a system that can be configured to generate either conservative versions that replicate an important portion of the inspiring set or innovative versions that depart from the inspiring set.

The analysis of fine-tuning in Colton et al. (2001) presents this concept as a property exhibited by some creative systems, inherent in their design, and indicative of a certain lack of general applicability. The mechanisms in ASPID and ASPERA described above provide a means for the user to control the degree to which the system will try to reproduce its inspiring set. In some way, the criteria proposed by Colton et al could be applied to obtain a measure of the extent to which these features affect the relationship between the inspiring set and the result.

13.3.4 Starting from Semantic Relations Between Words

Some poetry generation methods focus on producing poems that feature specific relations between words. A particular relation between words is used as input, and the construction procedure employed is designed to produce a poem somehow built around that relation. This usually involves resorting to particular rhetorical figures

that rely on the relation in question. The inclusion of the relation tends to induce in the reader an illusion of coherence that enhances the effect of the poem.

13.3.4.1 Poetry Generation Based on Semantic Relations

The two-tier approach to poem construction – whereby lines of verse are built independently in an initial process and then combined into stanzas during a later process – had already been employed in the PoeTryMe system (Gonçalo Oliveira, 2012), which considered as input a set of templates for lines (which came from actual lines from a corpus of poems) constructed so that each line template included gaps for two or more different words which had to be related to one another by a particular semantic relation. Constraints on content can be established by picking candidate fillers from a semantic relation graph – nodes are words, and edges are semantic relations between them – so that the selected words are in the relative vicinity of a given set of seed words. The actual poem is constructed by combining a set of filled-in line templates with a poem template that can establish constraints on the number of lines per stanza, the number of syllables per line, and/or the rhyme scheme for the stanza. The poem template is filled in according to one of a set of strategies. These strategies are based on searching for candidates that optimise a scoring function that considers satisfaction of the constraints of the poem template. The set of strategies includes generate-and-test and evolutionary methods.

The approach of Colton, Goodwin, and Veale (2012) uses as its main input knowledge a collection of similes obtained by a complex process of expanding an existing database of similes and then selecting from the result those that satisfy basic criteria for how correct and how easy to understand those similes are. The selected similes are taken together with a set of keywords extracted from newspaper articles. It is also based on template extraction, but it presents a much richer process of construction, involving up to three tiers of recombination, and it includes a very elaborate inventive process for building the initial seeds for the content. It also introduces emotional constraints in the form of a mood set at the start of the process – from an analysis of newspaper articles for a particular day – that influences later decisions. The form of the poems is constrained during a three-stage process: the selected words are used to build phrases, a first tier of templates is used to combine phrases into larger fragments of text, and a second tier of templates is used to build stanzas and/or poems from fragments of text.

Another system heavily influenced by semantic information used to drive the poetry generation process is described by Veale (2013). This system exploits poems as summarisation and visualisation devices for the set of properties and feelings that are evoked when a certain term T is presented in relation or in contrast to a related term M. That is, it explores the conceptual space of poems built around the metaphoric view of T as an M. It achieves this by relying on a rich semantic knowledge base mined from three resources: a large roster of stereotypes, a large body of normative relationships between these stereotypes, and the Google *n*-grams. The initial set of stereotypes is progressively elaborated – enriched with information

on relations between stereotypes and combined into complex conceptual blends – to provide semantic input pregnant with insight and wit. This material comes out as a set of pairings between elements from the domain of T and elements from the domain of M. It is exploited in poetic form by the use of a semantic grammar for the poem, which comes with gaps in particular lines for the two elements of a pair, and indications of the trope that should be used in different parts of the poem, how each line is to be rendered – as an assertion, an imperative, a request or a question – and whether it should be framed positively or negatively.

13.3.4.2 Assessing Outputs Obtained from Semantic Relations between Words

All poetry generation systems that have a construction procedure based on seeds in the form of semantic relations between words share similar characteristics in terms of the relation between their input knowledge and their results. Because the semantic inputs affect specific words, which are then presented as part of a larger template, the percentage of each phrase in the output that is predetermined in the input data is relatively large. This is not so apparent when one is perusing the output, because the construction procedure usually replaces key words in the template with words triggered by the semantic input. The resulting sentences are then structurally similar but apparently about altogether different things each time.

It might be interesting to consider whether a measure of similarity as proposed by Ritchie would capture this type of structural repetition inherent in the construction procedure for this particular approach.

13.3.5 Starting from Specific Emotions

Another approach to poetry generation is to target a specific emotion, so that the resulting poems somehow express the emotion in question.

13.3.5.1 Poetry Generation Targeting Specific Emotions

The approach by Misztal and Indurkhya (2014) relies on word-based specification of content and combines it with emotional aspects. It introduces a different way of constraining form and a different approach to the combination of modules dealing with different aspects of poem generation into a single system. This is a poetry generation system that relies on a multi-agent blackboard approach to create poems that employ a wide range of literary tropes and aim to convey a particular emotion. The system relies on a set of expert modules that each focus on a particular aspect, and which interact by sharing results on a blackboard. Types of experts include *word-generating experts*, which contribute words matching a given topic or emotion, *poem-making experts*, which arrange words from the common pool into phrases or

sentences guided by Context Free Grammars, and *evaluating experts*. This approach constrains the content in terms of a particular topic and an emotion extracted from the input text, and constrains the form through the implementation of the various experts. This type of solution allows fine-grained control of many aspects of the form, including, explicitly, many literary tropes. It is important to note that in this particular approach, specific literary forms are introduced explicitly by the set of system modules, rather than arising from the reuse of an existing corpus of poetic texts.

13.3.5.2 Assessing Outputs Obtained from Specific Emotions

Poetry generation systems that consider an emotional value as an additional input have the advantage of including an extra axis of specification. This may allow more variation in the set of results. However, the restrictions on relations between the inspiring set and those parts of the input knowledge that determine more closely the linguistic form of the outputs will hold in the same way as for the approaches discussed earlier.

13.3.6 Starting from Semantically Specified Content

The ultimate approach to poetry generation is to produce successful poems that match an input message that is semantically specified at the start. This corresponds to a refinement of the poetry generation task considered as a prose-to-verse conversion, where the desired message is expressed semantically instead of in prose. It has the advantage of not having to identify the meaning of the input message expressed as prose.

13.3.6.1 Poetry Generation Targeting Specific Semantic Content

H. M. Manurung (1999) relied on chart generation, taking as input a specification of the target semantics in first-order predicate logic, and a specification of the desired poetic form in terms of metre. Words are chosen from a lexicon that subsumes the input semantics, and a chart is produced incrementally to represent the set of possible results. At each stage, the partial solutions are checked semantically to ensure that no sentences incompatible with the original input are produced. Additionally, partial results are checked for compatibility with the desired poetic form. In this system, the process of grammar-based generation is driven by a semantic input, and the validation of the results relies on a poetic expert.

Manurung went on to develop in his PhD thesis (H. M. Manurung, 2003) an evolutionary solution for poetry generation – now described in (R. Manurung, Ritchie, & Thompson, 2012) – that was also aimed at a specific semantic target. Manurung's

MCGONAGALL used a linguistic representation based on Lexicalized Tree Adjoining Grammar (LTAG) over which operated several genetic operators – from baseline operators based on LTAG syntactic operations to heuristic semantic goal-directed operators – and two evaluation functions – one that measured how close the stress pattern of the solution was to the target metre, and one that measured how close the propositional semantics of the solution was to the target semantics. Both of these systems developed by Manurung included constraints on content – in terms of particular meaning to be conveyed, represented semantically – and constraints on form – formulated at the metric level and as requirements that outputs be syntactically correct with respect to a given grammar.

13.3.6.2 Assessing Outputs Obtained from Semantically Specified Content

The chart generation work of H. M. Manurung (1999) employs much more complex input knowledge, regarding syntax, semantics and metre. In this case it is not so clear what the inspiring set can be considered to be, because no poems are mentioned explicitly as being taken by the program as inspiration or input knowledge, but rather the restrictions of a particular poetic form are employed.

The MCGONAGALL system developed later by Manurung (H. M. Manurung, 2003; R. Manurung et al., 2012) is outstanding in that it does not build its input knowledge from a particular set of example poems, but rather operates with abstract descriptions of both linguistic correctness – considered in terms of an LTAG) for the language in question – and abstract descriptions of the poetic forms to be obtained – considered in terms of templates for sequences of feet, represented as skeletons for stanzas where only the strong and weak beats of the syllables are marked. In this particular sense, this is the system that least constrains the final nature of its output. The application of Ritchie's criteria to the MCGONAGALL system may prove difficult, as it is not clear what the inspiring set might be in this case. An analysis carried out in terms of the type of input knowledge it uses suggest it would be very difficult for elements of this hypothetical inspiring set to be replicated by this means. In fact, given this type of construction procedure, it is highly probable that even obtaining typical results may be difficult, because, in contrast to other systems, there is no feature in the input knowledge that constrains it to specific forms. The constraints on form are imposed by the fitness functions, and the task of finding successful candidates is delegated to the evolutionary search procedures. As the search space in question is significantly large, this may be a very difficult endeavour. The results reported in Manurung's published work suggest that the difficulty involved in this type of search is greater than might have been anticipated.

13.3.7 Evolutionary Approaches

Another preferred approach to poetry generation has been to consider evolutionary approaches where a population of candidate drafts is evolved over time by means of a number of operators until it achieves acceptable results under a given fitness function.

13.3.7.1 Poetry Generation Based on Evolutionary Procedures

Levy (2001) proposed a computer poet based on evolutionary computation, aided by an evaluation function implemented as a neural network trained on data obtained from human testers. The system had *generator modules*, which produce an initial population of candidate poems and modify it in succeeding generations, *evaluator modules*, which select the highest ranking individuals in each generation, a *work space*, in which the current population resides, a *lexicon*, a *conceptual knowledge base* and a *syntactical knowledge base*.

The Poevolve system operates on a representation of the lexicon that centers on the phonetic information. Evaluation is carried out by a neural network trained on judgements provided by a panel of experts on a single parameter – the likeability or creativity of a poem – on a scale of 1 to 6. The system generates randomly an initial population and allows it to evolve by applying a set of operations – mutation, crossover and direct copy – which in general terms are restricted to substituting one word for another. The results of the prototype were said to seem random in many ways, though there was an increase in value over the course of a program run.

The results of his earlier chart generation approach led Manurung to attempt an evolutionary solution (H. M. Manurung, 2003; H. Manurung, Ritchie, & Thompson, 2000b). This system draws on rich linguistic information (semantics, grammar) to generate a metrically constrained grammar-driven formulation of some given semantic content. The generation of poetry is attempted as the sequence of an initial transcription of the corresponding message into a semantic representation of its content, followed by the generation of a poem corresponding to that semantic representation. Given the intuition that there is a strong interaction between content and form during poetic composition by real people, this approach must surely lead to good modelling of the creative process. It has the disadvantage of being a knowledge-intensive approach to the problem, requiring strong formalisms for phonetics, grammar and semantics, together with some form of modelling of a certain aesthetic sense overlapping with all three.

13.3.7.2 Assessing Outputs Obtained from Evolutionary Approaches

The evaluation that has been applied to the WASP system might be extended to those systems following the evolutionary approach. However, certain differences between the two approaches must be taken into account.

Poevolve and WASP rely on different high level descriptions of the process of composition. Poevolve relies on a generate, evaluate, evolve cycle, whereas WASP follows a simple generate-and-test method. Of these, the description on which Poevolve is based is possibly a more accurate description of the actual creative process. Given enough information (about the elements that are manipulated and in such a form that the evaluators can take it into account) it does have great potential as claimed. However, it is not clear whether the amount (or rather the kind) of information required (syntactic, semantic) is as easily coded in terms of connectionist computing as the kind of information that the current prototype of Poevolve is using (mostly phonology and metrics).

On the other hand, the first prototype of Poevolve and WASP have in common the underlying assumption that consideration of phonology and metric with little regard for semantics and syntax does lead to reasonably 'poem-like' results. This may be related to the discussion by H. Manurung, Ritchie, and Thompson (2000a) regarding how automatically generated poetry is evaluated by humans with more leniency than the equivalent efforts in prose, and how this holds a danger of relaxing into easy simulations with little real merit.

The use of a neural network in Poevolve to evaluate the results could solve many of the problems faced when evaluating automatically generated poems. Nonetheless, the introduction of the training process of a neural net within the evaluation/feedback loop applied by the program introduces additional complexity into the already nebulous chain of valuations that take place when humans judge poetry.

Again, when defining evaluation methods for the evolutionary system of H. Manurung et al. (2000a, 2000b), it will be difficult to consider whether there is such a thing as an inspiring set, since the system seems to be working from general rules about poetry rather than specific examples of poems.

13.3.8 Starting from n-Gram Models of Language

Yet another approach to reusing text to constrain the output of poetry generators is the use of *n*-grams to model the probability of certain words following on from others. This corresponds to reusing fragments of the corpus of size *n*, and combining them into larger fragments based on the probability of the resulting sequence. This approach introduces a new approach to the generation of text, beyond template filling and grammar-based construction: the generation of text from an *n*-gram-based language model.

13.3.8.1 Poetry Generation Based on *n*-Gram Language Models

The Poetic Machine (Das & Gambäck, 2014) for generating Bengali poetry employs a line-based construction procedure, driven by a given rhyme pattern to be matched, with *n*-gram-based constraints used for selecting the final candidate for a line.

Combining *n*-gram modelling and evolutionary approaches, a redesigned version of the WASP poetry generator (Gervás, 2013a, 2013b) has been built using an evolutionary approach to model a poet's ability to iterate over a draft, applying successive modifications in search of a best fit, and including the ability to measure metric forms. It operates as a set of families of automatic experts that work in a coordinated manner like a cooperative society of readers/critics/editors/writers. These automatic experts together generate a population of drafts on which they all operate, modifying it and pruning it in an evolutionary manner over a number of generations of drafts, until a final version, the best-valued effort of the lot, is chosen. In this version, the overall style of the resulting poems is strongly determined by the accumulated sources used to train the content generators, which are mostly *n*-gram based. Several versions of this system have been developed, covering poetry generation using different inspirational sources where different sets of training corpora are used, ranging from a collection of classic Spanish poems (Gervás, 2013a) to a collection of news paper articles mined from the online edition of a Spanish daily newspaper (Gervás, 2013b).

A more refined attempt at generating poetry based on *n*-grams, including more abstract constraints in a tractable and complete search procedure, is the work of Barbieri, Pachet, Roy, and Esposti (2012). Relying on Constrained Markov Processes to generate texts in the style of the lyrics of an existing author, it integrates the constraints on grammaticality, rhyme, metre, and, to a certain extent, semantics into the search procedure itself. The system starts from a given word, and considers as semantically acceptable outputs those that include the *n* words in the corpus most closely related to the chosen word. This approach basically enriches an *n*-gram-based solution for text generation, building the constraints that drive the process into the construction procedure itself.

13.3.8.2 Assessing Outputs Obtained from *n*-Gram Models of Language

In poetry generation systems based on *n*-gram models of language, the input knowledge that is being considered is a large knowledge base that specifies how often particular sequences of words occur in a given language. Such knowledge bases are usually constructed from an analysis of a particular corpus of texts. In these cases, such corpora could in some sense be considered to be the inspiring set for the system. Construction procedures differ, but they are generally designed to ensure that the sequence of words in the resulting text is not totally improbable. A certain amount of adjustment is possible here. If sequences are forced always to be of the highest possible probability, the likelihood of replicating sentences in the original corpus will be very high. This effect may be reduced by choosing possible sequences but not the most probable ones. This usually guarantees outputs that are of reasonable typicality (human judges may accept them as valid utterances in the language) but do not necessarily replicate items in the inspiring set/corpus. Additional constraints on the construction procedure will have to be applied to ensure conformance to poetic form.

13.4 Applying Creativity Measurements to a Particular Example

The review of poetry generation systems presented in Section 13.3 covers many approaches and discusses how evaluation procedures might be applied to each of them and what issues may arise in evaluating different approaches. Ideally, the proposed evaluation methodologies should tested empirically on all the various systems. As it is impractical to report in a single book chapter, a particular example of a practical experiment in quantitative evaluation of the results from an automatic poet was chosen to be carried out.

Results suitable for carrying out this experiment results were available for the WASP, ASPID and ASPERA systems. In an initial analysis, it was noticed that the data collected for the original evaluations of WASP could be fitted to the scheme proposed by Ritchie. This represented a twofold advantage.

On one hand, data were available with no need for further evaluation processes. The evaluation of poems requires a set of volunteers to read through a set of results and produce a quantitative evaluation for each of the chosen parameters. Such an effort had been carried out for the WASP system (Gervás, 2000b). The resulting set of data, in the absence of a methodological framework suited for its analysis, had proved less productive than expected. As a result, the evaluation of subsequent attempts had been more focused on aspects directly relevant to specific design issues (Gervás, 2000a, 2001b).

On the other hand, there were further versions of the system that had evolved from the system evaluated. The new versions were designed based on a simple analysis of the results with no specific methodological framework. This meant that any conclusions obtained by applying evaluation frameworks could be compared with the conclusions found at the time. This should demonstrate whether the application of the proposed method is useful for drawing informative conclusions from raw data.

We therefore considered a generating program, WASP, which for the present purposes can be described as follows. The inspiring set was taken to be a specific 16th century Spanish classical sonnet. This established a number of restrictions on the poetry that was to be composed. Lines should have 11 syllables, according to very strict stress patterns. To simplify matters, the artefacts that the program aimed for were the simpler stanzas that make up a sonnet (two *cuartetos* – four lines each, rhyming ABBA ABBA – and two *tercetos* – three lines each, rhyming either ABA BAB or ABC ABC). As a first approximation, the generating program was set to attempt a *cuarteto* in isolation.

The construction process that was employed was designed to ensure that all resulting items had the correct syllable count and a valid pattern of stressed syllables for each line. Given a specific stanza to aim for, the system attempts to build an instance of this stanza based on the set of line patterns it received and the available vocabulary. Wherever several possible choices of words matched the metric constraints, the program made a random choice. This provided the non-determinism required to obtain multiple results on different runs. In each case, the final result might have reached the required number of lines or it might have stopped beforehand, unable to

meet the metric constraints on the remaining lines with the material available. The system was allowed a certain freedom in the following respects:

- it might or might not find rhymes between lines;
- it might or might not complete a full stanza;
- it might or might not achieve a syntactically correct poem.

13.4.1 Applying Ritchie's Criteria

Ritchie presented his criteria for assessing creativity (reviewed above in Section 13.2.1) based on the assumption that running the program produces a set of basic items, rather than a single item. In the present case, this could be simulated by running the program with the same parameters several times in order to produce a set of items.

13.4.1.1 Assessing Creativity Based on Existing Evaluation

As a first approach, the evaluation functions *typ* and *val* were defined informally in the following terms. A poem is considered *typical*, as encoded in the *typ* evaluation function, if it has the required number of lines and it has a syntactically correct reading. A poem is considered *good*, as encoded in the *val* evaluation function, if anything in it appeals to the aesthetic sense of the evaluator.

The *mapping function* that takes one from a particular combination of the initial data values to a specific set of results is given by the construction algorithm (while no full stanza has been achieved, find an appropriate line pattern, and fill that line pattern with adequate words).

The initialisation required to set this process in motion must provide the following information:

- *Alternatives for line patterns*: these are obtained from the lines in the sonnet used as the inspiring set, and each one corresponds to the sequence of POS tags corresponding to the words appearing in a line of the sonnet.
- *Alternatives for vocabulary*: these are obtained from the words of the original sonnet plus a number of additional words; each word carries additional information relating to the POS tag corresponding to it, the number of syllables, the position of its stress and word boundary information that affects the way it combines with neighbouring words to form the metre of the line.
- *Alternatives for structure*: in the present case, the types of stanza under consideration; a specific stanza must be chosen.

The set of initial data values are:

- a set of patterns;

- a given vocabulary;
- a specific stanza to aim for.

Each run of the program with such an initialisation produces either a complete stanza of the desired form or as many lines as can be produced while meeting the metric criteria. In order to obtain results that could be analysed according to Ritchie's framework, each set of 12 runs with the same initialisation was studied as a single set of results. Fourteen different initialisations were considered. This gave a total of 168 resulting poems. Each poem was evaluated by a team of volunteers, who were asked to provide two numerical values, one measuring the syntactic correctness of the poem (on a scale from 0 to 5) and one measuring the aesthetic qualities of the poem (on a similar scale). These values were combined with the number of lines in each poem to provide an approximation to the two evaluation functions required (as described above).

The resulting values for these results under Ritchie's criteria are presented in Table 13.3. Table 13.4 presents the parameters that were employed to construct the results in Table 13.3.

Table 13.3 Results for 14 criteria

Criterion 1	Average typicality	0.71
Criterion 2	Typical results/results	0.54
Criterion 3	Average quality	0.47
Criterion 4	Good results/results	0.24
Criterion 5	Good typical results/typical results	0.36
Criterion 6	Good atypical results/results	0.05
Criterion 7	Good atypical results/atypical results	0.12
Criterion 8	Good atypical results/good typical results	0.28
Criterion 9	Results in the inspiring set/inspiring set	0.00
Criterion 10	Results/results in the inspiring set	∞
Criterion 11	Average typicality new results	0.71
Criterion 12	Average quality new results	0.47
Criterion 13	Typical new results/results	0.54
Criterion 14	Good new results/results	0.24

Table 13.4 Basic parameters employed in the first approach

Weight for poem length	0.5
Weight for syntactic correctness	0.5
Typicality threshold	0.7
Quality threshold	0.7
Total number of results	168
Number of items in the inspiring set	2

Only the first eight criteria are relevant, because none of the inspiring set reappears in the result. This is apparent in the fact that the original formulation of criterion 10 tends to infinity as the number of results already present in the inspiring set tends

to 0.[4] This is due to the fact that the construction process actually first factorises and then recombines elements of the inspiring set, adding additional words from the vocabulary. This reduces greatly the probability that an element in the inspiring set will be generated anew by the system. An immediate consequence is that criterion 9 drops to zero and criterion 10 rises to infinity. Additionally, those criteria designed to capture specific differences between items that are new and items already in the inspiring set produce the same score as the original criteria they were evolved from (the same values result for criteria 11 and 1, 12 and 3, 13 and 2, and 14 and 4). Finally, because all items produced are novel items, some of the values for the new criteria proposed by Ritchie (2007) that concern proportions with respect to the novel items produce the same score as the original criteria they were evolved from (the same values result for criteria 15 and 2, and 16 and 4). The new criteria 17 and 18 are similar to the original 6 and 7 but differ in their formulation regarding typicality.[5]

A question that may need detailed discussion is how one identifies whether an element in the inspiring set has reappeared in the results. For this version of the system, none of the *cuartetos* in the inspiring set appears as such among the results, but some of the lines of the poems in the inspiring set may reappear, and – given the construction procedure employed – all of the lines in the results will have a syntactic structure that is borrowed from the lines in the inspiring set.

The system is better at producing typical items than at producing good items (the score is higher for criterion 1 than for criterion 3 and higher for criterion 2 than for criterion 4). This makes sense, since all decisions about the system (algorithms applied and constraints imposed) during the construction process are concerned with ensuring the production of typical items, rather than good ones. In fact, the system has no means of identifying good items, and therefore cannot be expected to aim towards them during construction.

Atypical results score badly in terms of quality. This may be due to the evaluators not having a clear idea of whether their judgement about the quality should take into account how typical the item was. Th evaluators might be awarding good scores for quality to items that were typical. This would imply that their own reaction was to apply criterion 5 rather than criterion 4. The fact that the system performs better under criterion 5 than criterion 4 with these evaluators may be taken as evidence in favour of this interpretation. Criterion 8 provides an indication of this relation (low presence of atypical results among the good results).

13.4.1.2 Effect of Evaluation Parameters on Creativity Assessment

The data presented so far are based on a specific selection of parameters to be employed during evaluation. The value obtained for *typ* is actually the result of combining mathematically the values assigned for syntactic correctness and the

[4] It has proven impossible to recover the original data to recalculate the value for the criterion according to its new formulation.

[5] Unfortunately, this means that no values can be produced for them without access to the original data from the experiment, which are no longer available.

number of lines obtained for each attempted instance of the stanza. The actual formula applied to obtain the final value corresponds to what Ritchie calls a *weighted property rating scheme*, as used for evaluating typicality. The role of the weights employed in the actual combination needs to be discussed.

Additionally, two threshold values were applied to distinguish highly rated items on typicality or quality.

The present section considers whether alternative assignments of values to these parameters affect in any significant way the conclusions drawn from the results. Criteria 9 to 14 have been omitted from the discussion, since they play no significant role.

The first decision that may affect the evaluation is the relative importance that number of lines and syntactic correctness play in our assessment of typicality. The evaluation discussed above considers them with equal relative importance: weight assignment for syntactic correctness = 0.5 and weight assignment for number of lines = 0.5. Table 13.5 considers a similar evaluation but including two new different alternative weight assignments:

- alternative A0 (the original one with weight assignment for syntactic correctness = 0.5 and weight assignment for number of lines = 0.5);
- alternative A1 (weight assignment for syntactic correctness = 0.7 and weight assignment for number of lines = 0.3); and
- alternative A2 (weight assignment for syntactic correctness = 0.3 and weight assignment for number of lines = 0.7).

Comparatively, alternative A1 gives more importance to syntactic correctness, alternative A0 gives them equal importance, and alternative A2 gives more importance to the number of lines.

Table 13.5 Different weightings for typicality

Criterion	Description	A0	A1	A2
1	Average typicality	0.71	0.67	0.75
2	Typical results/results	0.54	0.48	0.79
3	Average quality	0.47	0.47	0.47
4	Good results/results	0.24	0.24	0.24
5	Good typical results/typical results	0.36	0.34	0.29
6	Good atypical results/results	0.05	0.08	0.01
7	Good atypical results/atypical results	0.12	0.16	0.06
8	Good atypical results/good typical results	0.28	0.52	0.05

The results show that the average typicality (criterion 1) drops for alternative A1 and rises for alternative A2. This is due to the fact that the constraints applied during construction take only the number of lines of the stanza into account explicitly, but correct syntax is considered implicitly in the reuse of line patterns. This effect is mirrored in criterion 2 (which represents the same information but relative to the total set of results) and it does not affect criteria 3 and 4, which focus on quality rather than typicality. For criteria 6 to 8, which focus on atypicality, the effect is observed in

reverse (values rise for alternative A1 and drop for alternative A2). Criterion 5, which measures the ratio of good results within the set of typical results, shows a different behaviour, with values dropping from alternative A0 to A1 to A2. As the changes in weighting affect the value for typicality, which basically establishes the set of candidate artefacts being considered for this criterion, the observed results suggest that a higher number of good results appear in the set of candidates determined when using alternative A0. This matches the intuition that alternatives biased towards one or the other of the component functions tend to include more of the less-valued candidates.

Another possible way of affecting the evaluation is to vary the thresholds that are used to distinguish highly rated items in each class (typical or good). Criteria 1 and 3 are not affected by this change, since they do not refer to the threshold value. Therefore they are omitted from the following discussion.

The threshold value for quality determines how many items are considered good, and therefore affects criteria 4 to 8. The threshold value for typicality affects criterion 2 and criteria 5 to 8.

In the result sets discussed below the weight assignment for typicality was maintained at syntactic correctness = 0.5 and number of lines = 0.5 – the same as for the initial discussion.

Table 13.6 shows the values for the relevant criteria for the same set of 168 generated poems with five different threshold combinations for quality and typicality:

- A: equal high thresholds;
- B: equal medium thresholds;
- C: equal low thresholds;
- D: high typicality and low quality thresholds; and
- E: low typicality and high quality thresholds.

Table 13.6 Equal high thresholds

Criterion	A	B	C	D	E
2	0.54	0.88	0.89	0.54	0.89
4	0.24	0.50	0.68	0.68	0.24
5	0.36	0.57	0.77	0.89	0.28
6	0.05	0.00	0.00	0.21	0.00
7	0.12	0.00	0.00	0.45	0.00
8	0.28	0.00	0.00	0.44	0.00

It can be seen from the results that lowering the typicality threshold results in a zero score for criteria 6 to 8. This is because these criteria involve good atypical results. When the typicality threshold is lowered the number of atypical items is reduced, and any reduction brings down the number of good items to be found among them. Criteria 2 (regarding typicality) and 4 (concerned with quality) are inversely proportional to the threshold applied in each case – the value for the corresponding criterion falls when the threshold rises and rises when it falls. Criterion 5 is different in every case because it involves both thresholds.

There is great variation in the values obtained for each of these criteria when the thresholds are moved. This implies that the assignment of specific values for these thresholds should be done beforehand based on domain-specific criteria, or oriented towards the specific aims that have been established for the system.

An additional alternative is to consider different thresholds for distinguishing typical and atypical items. So far, items that did not rate highly on typicality have been considered atypical. A finer-grained approach would establish a low threshold below which items would be considered as atypical. This might establish a high threshold value to determine when an item is typical and a low threshold value to determine when an item is atypical.

13.4.1.3 Evaluation of the Degree of Fine Tuning

The criteria defined by Colton et al. for measuring fine-tuning (reviewed above in Section 13.2.2) are somewhat difficult to apply to this case for the following reasons:

- It is difficult to isolate particular items of knowledge from the rest, since the contribution for a given line pattern, for instance, is tightly coupled to the existence of word items in the necessary category.
- The same words may be coupled with different line patterns if they belong to a category which appears in more than one line pattern.

For simplicity, the whole set of knowledge employed is considered as K to obtain a first approximation of the criteria.

In the case under discussion, there are no reinventions. This means that for the particular set of input knowledge employed, $C_K = O_K$. With respect to the criteria described for measuring the degree of fine tuning $ft(K)$ yields a value of 0 – there are no reinventions among the dependency set for the knowledge K employed.

This measure can be misleading, and this is probably related to having applied too strict an interpretation of what it takes for an artefact to be considered as part of R_K. In their description of their measurements, Colton et al. (2001) consider only whether an artefact in the output set is exactly the same as one in the inspiring set. This is adequate for the mathematical domain (in which the discussion in that paper is mostly based), where the elements being generated are structurally simple and the probability of replicating the ones in the inspiring set is high. In the domain of poetry, as soon as the conceptual unit of the poem is broken down into its constituent elements and the construction process is allowed to recombine these elements in different ways, the probability of reproducing a poem originally in the inspiring set is very low.

In his revision of his original paper, Ritchie (2007) addresses the fact that the original formulation of the criteria is weakened by poor handling of the notion of similarity. All of the discussions of novelty rely on an artefact in the inspiring set being replicated identically in the results. Although this is a sufficient condition for the result not to be considered novel, it is not a necessary one. Results that are similar

beyond a certain threshold to elements in the inspiring set may also give rise to a judgement of non-novelty.

For these cases, it may be more fruitful to employ some measure of similarity between the artefacts in the output set and those in the inspiring set to decide which artefacts are to be considered as reinventions. Artefacts that are very similar to those in the inspiring set, even if they are different, should be considered as reinventions. A creative program that produces artefacts only marginally different from those in the inspiring set should probably be considered to be using fine-tuned knowledge.

In this sense, the criteria proposed for measuring fine-tuning should be extended in the case of artefact domains of higher complexity, possibly taking into account measures akin to those proposed in Pease, Winterstein, and Colton (2001).

13.4.2 Analysis of the Results

The type of evaluation employed here seems to fall short in terms of identifying the real value of the resulting poems. This may be due to unforeseen assumptions made by the human evaluators about what is considered novelty and quality in this context. We consider that typical *cuartetos* have four lines. Results with fewer than four lines are considered atypical – the shorter they are, the more atypical.

Typical *cuartetos* in 16th-century Spanish poetry generally fulfil the following conditions:

- They include striking vocabulary items (words still in use today but used at the time with different meaning, words no longer used, words referring to objects or concepts that date specifically from that time).
- Their grammar is sometimes difficult to understand (owing to obsolete turns of phrase or the use of hyperbaton, a poetic ornament which relies on shuffling the elements of a sentence in ungrammatical ways in order to satisfy metric constraints or achieve poetic effects).
- They seldom occur in isolation, so they are rarely self-contained units from a syntactic or semantic point of view.

These issues may have played a role in making evaluators rate some of the resulting poems highly: the presence of striking vocabulary items, obscure grammar and attribution of a hypothetical context in which certain turns of phrase might make sense. However, they are not necessarily desirable features in poems in a wider domain.

In view of this conclusion, a more restricted method of evaluation should be defined to represent more closely the ingredients at play in this domain. To counter the effect described, such a method of evaluation should provide evaluators with explicit guidelines on how to rate the poems, and which features to take into account when doing so. For instance, it might be considered that a *cuarteto* is good depending on the following aspects:

- Rhyme (very good if it rhymes ABBA, good if all verses rhyme in some way, acceptable if some verses do not rhyme and bad if none of the verses rhyme).
- Syntax (very good if it is a self-contained syntactically correct unit, good if it can be parsed as a syntactically correct fragment within some hypothetical context, acceptable if it does not contain any striking syntactical errors and bad if it does).
- Poetic ornaments (the quality of a poem rises if it contains any combination of words that can be interpreted as a poetic ornament).

13.5 Conclusions

The evaluation of automatically generated poetry still requires a lot of work. Existing theoretical proposals on how to evaluate creativity can be applied to this domain, but fall short in various ways owing to the intrinsic complexity of the artefacts being generated and the kind of input knowledge that is needed.

In general terms, the aspects that need evaluation can be summarised as follows. Artefacts must at the same time be capable of:

- meeting a subset M of the requirements of the evaluation function (met requirements);
- challenging a subset C of the requirements of the evaluation function (challenged requirements).

An existing set of artefacts of a given kind allows the prediction of subsequent artefacts of the same kind, with a given probability. Items that are predictable in this sense are not considered creative. Items that cannot be adequately linked to the preceding sequence are not considered creative.

This issue must be discussed relative to the particular domain to which the items belong. For conservative domains, typical items tend to be good, and good items tend to be typical. This approach gives rise to a somewhat stilted style. For innovative domains, atypical items tend to be good, and typical items are bad. The resulting style is more dynamic.

In relation to Boden's distinction between exploratory and transformational creativity, conservative domains rely more on exploratory creativity and innovative domains rely more on transformational creativity.

For the particular field of poetry, as considered in existing attempts at automatic generation, the identification of the domain in the sense used here relates more to the poetic form that is being aimed for. Certain poetic forms are more conservative, and others are more innovative.

Regarding the issue of measuring the extent to which a given creative program has been fine-tuned to produce a particular type of item, the experiments reported in this chapter have shown shortcomings in the available definitions. Existing descriptions of how this measurement may be achieved need to be extended to take into account domains where artefacts that are too similar to those in the inspiring set may be considered just as unoriginal as the items in the inspiring set. Such is the case in the

poetry domain discussed in this chapter. On the other hand, measuring fine tuning may not be enough to give an idea of how adequate a creative program is, in the sense that an equivalent measure may be required to capture possible inadequacies at the other extreme: programs that are unable to reproduce any of their inspiring set.

Acknowledgements The original version of this chapter was published as Gervás (2002). To be included in the present compilation, it has been revised in several ways. The original review of poetry generation system has been replaced with an up-to-date survey of the systems that have been developed since then. The descriptions of the various research efforts mentioned in the paper have been extended to include later work by the authors. The part of the original discussion section that addressed the applicability of evaluation methods to particular approaches to poetry generation has been shifted forward to be integrated with the review of the generation approaches. The analysis of the empirical data originally collected has been expanded where possible to cover new considerations arising from the revised version of the evaluation criteria. As mentioned in the original paper, I would like to thank the group of volunteers who carried out the task of assigning real values to the set of results - "poems" - considered in this chapter, and to thank the referees for their comments which helped improve this chapter.

References

Aamodt, A., & Plaza, E. (1994). Case-based reasoning: Foundational issues, methodological variations, and system approaches. *AI Communications*, *7*(1), 39–59.

Agirrezabal, M., Arrieta, B., Hulden, M., & Astigarraga, A. (2013). POS-tag based poetry generation with WordNet. In *Workshop on Natural Language Generation (ACL 2013)*, Sofia, Bulgaria.

Barbieri, G., Pachet, F., Roy, P., & Esposti, M. D. (2012). Markov constraints for generating lyrics with style. In *Proceedings of the 20th European Conference on Artificial Intelligence, ECAI 2012* (pp. 115–120). Frontiers in Artificial Intelligence and Applications. IOS Press.

Boden, M. (1990). *Creative Mind: Myths and Mechanisms*. London: Weidenfeld & Nicholson.

Charnley, J., Colton, S., & Llano, M. T. (2014). The FloWr framework: Automated flowchart construction, optimisation and alteration for creative systems. In *Proceedings of the 5th International Conference on Computational Creativity, ICCC 2014*, Ljubljana.

Colton, S., Goodwin, J., & Veale, T. (2012). Full-FACE poetry generation. In *Proceedings of the International Conference on Computational Creativity 2012* (pp. 95–102). Dublin.

Colton, S., Pease, A., & Ritchie, G. (2001). The effect of input knowledge on creativity. In *Proceedings of the ICCBR-01 Workshop on Creative Systems*, Vancouver, Canada: Technical Reports of the Navy Center for Applied Research in Artificial Intelligence.

Das, A., & Gambäck, B. (2014). Poetic machine: Computational creativity for automatic poetry generation in Bengali. In *Proceedings of the 5th International Conference on Computational Creativity, ICCC 2014*, Ljubljana.

Gervás, P. (2000a). Un modelo computacional para la generación automática de poesía formal en castellano. *Procesamiento de Lenguaje Natural*, (26), 19–26.

Gervás, P. (2000b). WASP: Evaluation of different strategies for the automatic generation of Spanish verse. In *Symposium on Creative & Cultural Aspects and Applications of AI & Cognitive Science* (pp. 93–100). University of Birmingham, United Kingdom.

Gervás, P. (2001a). An expert system for the composition of formal Spanish poetry. *Journal of Knowledge-Based Systems*, *14*(3–4), 181–188.

Gervás, P. (2001b). Generating poetry from a prose text: Creativity versus faithfulness. In *AISB 2001 Symposium on Artificial Intelligence and Creativity in Arts and Science*, University of York, UK: Society for the Study of Artificial intelligence and Simulation of Behaviour.

Gervás, P. (2002). Exploring quantitative evaluations of the creativity of automatic poets. In *Workshop on Creative Systems, Approaches to Creativity in Artificial Intelligence and Cognitive Science, 15th European Conference on Artificial Intelligence*, Lyon, France.

Gervás, P. (2013a). Computational modelling of poetry generation. In *Proceedings of the AISB'13 Symposium on Artificial Intelligence and Poetry*, Exeter, UK.

Gervás, P. (2013b). Evolutionary elaboration of daily news as a poetic stanza. In *Proceedings of the IX Congreso Español de Metaheurísticas, Algoritmos Evolutivos y Bioinspirados, MAEB 2013*, Madrid.

Gonçalo Oliveira, H. (2012). PoeTryMe: A versatile platform for poetry generation. In *Proceedings of the ECAI 2012 Workshop on Computational Creativity, Concept Invention, and General Intelligence*. C3GI 2012, Montpellier.

Jordanous, A. (2019). Evaluating evaluation: Assessing progress and practices in computational creativity research. In T. Veale & F. A. Cardoso (Eds.), *Computational creativity: The philosophy and engineering of autonomously creative systems* (pp. 209–234). Springer.

Levy, R. (2001). A computational model of poetic creativity with neural network as measure of adaptive fitness. In *Proceedings of the ICCBR-01 Workshop on Creative Systems*, Vancouver, Canada: Technical Reports of the Navy Center for Applied Research in Artificial Intelligence.

Manurung, H. M. (1999). Chart generation of rhythm-patterned text. In *Proceedings of the First International Workshop on Literature in Cognition and Computers*, Tokyo.

Manurung, H. M. (2003). *An Evolutionary Algorithm Approach to Poetry Generation* (PhD. Thesis, University of Edinburgh, Edinburgh, UK).

Manurung, H., Ritchie, G., & Thompson, H. (2000a). A flexible integrated architecture for generating poetic texts. In *Proceedings of the First International Workshop on Literature in Cognition and Computers.* Chiang Mai, Thailand.

Manurung, H., Ritchie, G., & Thompson, H. (2000b). Towards a computational model of poetry generation. In *Symposium on Creative & Cultural Aspects and Applications of AI & Cognitive Science*, University of Birmingham, UK.

Manurung, R., Ritchie, G., & Thompson, H. (2012). Using genetic algorithms to create meaningful poetic text. *Journal of Experimental and Theoretical Artificial Intelligence*, *24*(1), 43–64.

Misztal, J., & Indurkhya, B. (2014). Poetry generation system with an emotional personality. In *Proceedings of the 5th International Conference on Computational Creativity, ICCC 2014*, Ljubljana.

Oulipo. (1988). *Atlas de littérature potentielle*. Folio: Essais. Gallimard.

Pease, A., Winterstein, D., & Colton, S. (2001). Evaluating machine creativity. In *Proceedings of the ICCBR-01 Workshop on Creative Systems*, Vancouver: Technical Reports of the Navy Center for Applied Research in Artificial Intelligence.

Queneau, R. (1961). *100.000.000.000.000 de poèmes*. Gallimard Series. Schoenhof's Foreign Books.

Rashel, F., & Manurung, R. (2014). Pemuisi: A constraint satisfaction-based generator of topical Indonesian poetry. In *Proceedings of the 5th International Conference on Computational Creativity, ICCC 2014*, Ljubljana.

Ritchie, G. (2001). Assessing creativity. In *AISB 2001 Symposium on Artificial Intelligence and Creativity in Arts and Science*, University of York, UK.

Ritchie, G. (2007). Some empirical criteria for attributing creativity to a computer program. *Minds & Machines*, *17*, 67–99.

Ritchie, G. (2019). The evaluation of creative systems. In T. Veale & F. A. Cardoso (Eds.), *Computational creativity: The philosophy and engineering of autonomously creative systems* (pp. 157–192). Springer.

Toivanen, J. M., Gross, O., & Toivonen, H. (2014). The officer is taller than you, who race yourself! using document specific word associations in poetry generation. In *Proceedings of the 5th International Conference on Computational Creativity, ICCC 2014*, Ljubljana.

Toivanen, J. M., Järvisalo, M., & Toivonen, H. (2013). Harnessing constraint programming for poetry composition. In *Proceedings of the International Conference on Computational Creativity 2013* (pp. 160–167). Sydney.

Toivanen, J. M., Toivonen, H., Valitutti, A., & Gross, O. (2012). Corpus-based generation of content and form in poetry. In *Proceedings of the International Conference on Computational Creativity 2012* (pp. 175–179). Dublin.

Veale, T. (2013). Less rhyme, more reason: Knowledge-based poetry generation with feeling, insight and wit. In *Proceedings of the International Conference on Computational Creativity 2013* (pp. 152–159). Sydney.

Chapter 14
Multi-agent-based Models of Social Creativity

Rob Saunders

Abstract This chapter provides an introduction to the computational modelling of social creativity using multi-agent systems. It reviews motivations for modelling socio-cultural aspects of creativity computationally and describes a systems view of creativity that has influenced approaches to the computational of modelling social creativity. A minimal model of an 'artificial creative system' is described and the components of an individual agent are given in some detail. The Digital Clockwork Muse is presented as an implementation of an artificial creative system together with results from some small-scale investigations into the self-organisation of creative fields. Extensions of the computational model are described, including the evolution of domain-specific languages, more sophisticated individuals and alternative models of inter-agent interactions.

14.1 Introduction

Popular definitions of creativity maintain a distinction between personal and social creativity; Boden (1990) defines both psychological creativity (*p-creativity*) and historical creativity (*h-creativity*), while Gardner (1993) distinguishes between *little-c* (mundane) and *big-C* (eminent) creativity. These definitions maintain that creativity has two important but distinct meanings: a person's perception of their own work and an honorific title awarded by society. Models of creativity that attempt to reconcile these different meanings are complicated by the fact that an individual's creativity is productively connected with culture and learning (Lindqvist, 2003).

Computational creativity has attempted to address questions around the social nature of creativity in one of two ways, either (1) by producing highly sophisticated models of individual creativity that can interact with society at large, or (2) by devel-

Rob Saunders
The MetaMakers Institute, Falmouth University, Cornwall, UK
e-mail: rob.saunders@falmouth.ac.uk

T. Veale, F. A. Cardoso (eds.), *Computational Creativity*, Computational Synthesis and Creative Systems, https://doi.org/10.1007/978-3-319-43610-4_14

oping computational models of artificial societies that exhibit recognisable features of social creativity. The first approach requires the development of highly capable computational systems that not only are able to produce creative works, as judged by the standards of human experts, but are also capable social actors, for example, by conjuring up a persona for the computational system such that it may achieve some recognition of autonomy (Colton, 2012; Cope, 2005; Hoffman & Weinberg, 2011; McCorduck, 1991; Wiggins, 2008). The second approach requires the development of computational models of salient social and cultural aspects of creativity, for example, by developing multi-agent-based models of creative individuals (Bown & Wiggins, 2005; Gabora, 1995; Macedo & Cardoso, 2001; Saunders & Gero, 2001; Saunders & Grace, 2008; Sosa & Gero, 2005). This chapter explores the second approach.

Computational models of social creativity can be applied in different ways: as a way to understand creativity as a complex social phenomenon, similar to computational social science (Saunders & Bown, 2015); as a practical approach to developing distributed computational creativity systems, similar to other applications of multi-agent systems for distributed computing (Wooldridge, 2001); or as a way to support human creativity as a social activity (Saunders, Chee, & Gemeinboeck, 2013). This chapter focuses on the first of these applications, although the approach and techniques can be applied to the development of distributed systems for practical applications.

Cellular automata and agent-based models are well established in the synthetic study of social phenomena (Axelrod, 1997; Epstein & Axtell, 1996; Schelling, 1969; Wooldridge, 2001). In computational creativity, the study of creativity through the development of agent-based models has spanned multiple domains and different aspects of creativity as a social phenomenon. For example, Gabora's 'Meme and Variations' (MAV) was one of the earliest multi-agent-based models to examine the interactions between individuals that drive social creativity and cultural evolution through imitation and mutation of ideas (Gabora, 1995). Colton, Bundy, and Walsh (2000) developed a computational model involving multiple agents working together to explore a mathematical domain. Macedo and Cardoso (2001) explored the ability of agents to gain the attention of others through the production of 'surprising' artefacts. Sosa and Gero (2005) used multi-agent-based models to examine the role of society in design. Bown (2008) developed multi-agent-based models to explore cohesion, competition and maladaptation in the evolution of musical behaviour. In these agent-based models, creativity can be subjective or objective, individual or collective, direct or indirect (Sosa & Gero, 2008). This chapter presents a particular multi-agent model based approach that combines computational models of personal and social creativity to produce 'artificial creative systems'.

The next section briefly describes a systems view of creativity that provides a useful framework for developing computational models of social creativity. Section 14.3 presents a multi-agent approach to computational modelling based on the systems view of creativity. Section 14.4 describes an implementation of an artificial creative system, the Digital Clockwork Muse, and presents some results from experiments with this implementation. Section 14.5 discusses possible extensions to the

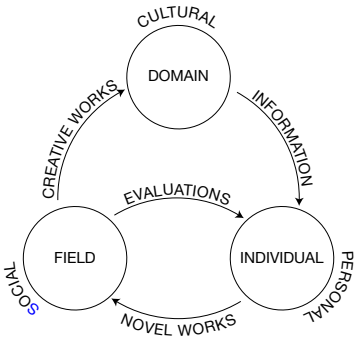


Fig. 14.1 Csikszentmihalyi's systems view of creativity. Based on an illustration from Saunders and Grace (2008), adapted from Csikszentmihalyi (1988).

model presented, including the evolution of domain-specific languages, alternative individuals agents and interactions between agents.

14.2 A Systems View of Creativity

Vygotsky (1971) first proposed a systems theory of creativity, emphasising a reciprocal relationship between individuals and their socio-cultural environment where individuals are influenced by their understanding of their socio-cultural environment and, through their actions, cause it to change. An individual may determine that their work is 'creative' independent of the judgement of others, but their determination is naturally informed by their experiences of their socio-cultural environment, for example, the work of others (Martindale, Moore, & West, 1988; Tardif & Sternberg, 1988). In addition, for a work to be given the honorific title of 'creative', other members of a society must agree based on their own experiences of the socio-cultural environment. Consequently, regardless of whether creativity is personal or social, it is the result of a dynamic system of interactions between multiple individuals and their socio-cultural environment (Engeström, 1996). Csikszentmihalyi (1988) argued that creativity is the product of three shaping forces that result from the cultural, social and personal context of the creative activity. The resulting model, illustrated in Fig. 14.1, defines creativity as the result of the interaction between three subsystems: a *domain*, an *individual,* and a *field*. Each subsystem performs a specific function; the domain transmits information to the individual, the individual produces a variation and the field selects variations to pass on to the domain. These subsystems are described in more detail below.

Csikszentmihalyi (1999) argues that before an individual can produce a variation there must already exist a culture, with traditions and conventions in place for the individual to draw on. A *domain* is defined as the body of knowledge, the set of rules and procedures, the symbolic system, which is used by an individual to

produce variations. There will be a multitude of domains in a culture (Feldman, Csikszentmihalyi, & Gardner, 1994), and domains will evolve and change over time. An individual must reference a domain to produce a contribution that a field will understand. As Boden (1990) points out: 'To be appreciated as creative, a work of art or a scientific theory has to be understood in a specific relation to what preceded it'.

An *individual* is the producer of variation within the systems model. Csikszentmihalyi argues that a person's background, personal traits and motivations, together with their ability to internalise domain knowledge as well as the expectations of the field, combine to enable an individual to be successfully creative within the system. This view of an individual emphasises the need for them to learn and adapt in order to gain a mastery of a domain and anticipate the response of a field to proposed variations. It also emphasises the importance of successful communication for a creative individual.

A *field* is composed of all of the individuals in a society who possess domain knowledge and have influence over its contents. Sawyer (2012) defines a field as 'a complex network of experts, with varying expertise, status, and power'. Possible members of a field include creators, educators, critics, agents (marketers) and consumers. According to Csikszentmihalyi, if the members of a field judge a contribution from an individual to be creative it will be added to the domain for other individuals to reference, thus continuing the cycle.

Csikszentmihalyi argues that creativity can be found when and where these three subsystems interact. In this view, an individual is necessary but not sufficient for a creative system; all three subsystems, and their interactions, are equally important. For example, Csikszentmihalyi (1988) argues that highly structured domains, for example, mathematics, promote creativity by assisting individuals accessing relevant knowledge and supporting fields assessing an individual's contribution. Similarly, social structures, for example, the emergence of 'gatekeepers' (Feldman et al., 1994; Sosa & Gero, 2004), have a significant impact on an individual's ability to have variations accepted. To emphasise the importance of all of the three elements and their interactions the model is often referred to as the Domain–Individual–Field–Interaction (DIFI) framework (Feldman et al., 1994).

14.2.1 Modelling the Systems View of Creativity Computationally

Liu (2000) first proposed an approach to computational modelling of the DIFI framework: the *dual generate-and-test* model of creativity, illustrated in Fig. 14.2. This encapsulates two generate-and-test cycles: one at the level of the individual and the other at the level of society. The domain is modelled as a repository of artefacts for the individual to draw on and for the field to contribute to. The individual generate-and-test cycle implements Newell et al.'s generate-and-test model of creative thinking (Newell, Shaw, & Simon, 1962) incorporating problem finding, artefact generation and personal creativity (p-creativity) testing. The socio-cultural generate-and-test cycle incorporates the individual as the generator and the field as a monolithic test

of the creativity of the variations generated, which determines whether they are sufficiently creative to be added to the domain. The apparent simplicity of Liu's computational model masks the complexity of modelling the field as a monolithic socio-cultural creativity test. Liu suggests that such a system would likely defer to some form of oracle, most likely a human, to provide judgements of creativity. The following section explores the use of multi-agent-based models to develop 'artificial creative systems' that support emergent notions of creativity from the interactions of individuals, removing the need for such oracles.

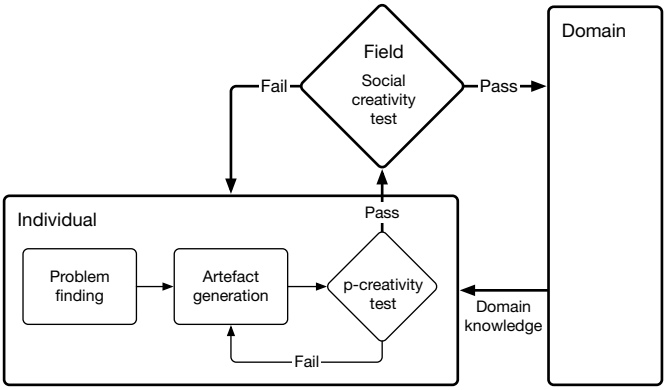


Fig. 14.2 The individual and socio-cultural generate-and-test loops in Liu's dual generate-and-test model of creativity. Based on an illustration in Saunders (2002).

14.3 Artificial Creative Systems

The agent-based model presented here does not attempt to define a 1-to-1 mapping with the subsystems of the DIFI framework. In an artificial creative system, no individual agent contains a test for 'big-C' or 'h-creativity' but instead tests for 'little-c' or 'p-creativity', which may or may not be judged by the other agents within a field to be 'creative'. The ability of agents to make independent judgements of both novelty and value is fundamental to the model, permitting the emergence of social definitions of creativity as a collective function of many individual evaluations of creativity. Consequently, the notion of a 'creative work', or a 'creative individual', is honorific, as it must be determined as the consequence of some form of negotiation between at least two agents.

The autonomy of agents equipped with the ability to determine what is interesting, and therefore potentially p-creative, is the key to adapting Liu's dual generate-and-test model to the study of emergent notions of creativity. This approach substitutes the monolithic socio-cultural creativity test with one based on a distributed agreement

that emerges from communication between individual agents. In such an artificial creative system, the socio-cultural creativity test is modelled through the communication of artefacts and evaluations of p-creativity between individuals. The following describes how the subsystems of the DIFI framework can be modelled as agents and interactions.

14.3.1 Domains

Creative domains, as described by Csikszentmihalyi (1988), are dynamically maintained and contain symbolic material, for example, rules and language, as well archive material, for example, previous works. Consequently, domains should be considered as being distributed across creative fields, existing within a variety of media, with each individual in a field having a partial view of the whole. The simplest computational model of a domain is a repository of artefacts that have been judged to be creative, i.e. an archive. This is the model described in Liu's model above and used in the simple implementations below. But this model lacks both the distributed and the multifaceted nature of the domain described by Csikszentmihalyi.

Other computational models of a domain are possible that capture better the distributed nature of the domain, for example, Gabora's MAV, where each agent maintains some part of the whole domain in memory. These partial views may overlap, i.e. two agents may have artefacts in common. In these models it is only by considering the intersection of these partial views, i.e. parts that are commonly held, that the domain can be understood. Saunders (2011) proposed a way to extend the model of a domain to encompass more than exemplars of previous work, by incorporating a model of the evolution of domain-specific languages to capture symbolic representations, for example, descriptions of artefacts.

14.3.2 Fields

In an artificial creative system, a field is modelled as a set of agents that interact with each other and the domain according to a communication policy, as illustrated in Fig. 14.3. In this example, agent i communicates an artefact that it considers to be p-creative to agent j, which evaluates the artefact according to its own p-creativity test and sends its evaluation back to i. Each agent's evaluation of an artefact is affected by the traits of the individual, for example, preference for novelty or prototypes held in memory. Consequently, through the communication of evaluations, j can affect the generation of future artefacts by i by rewarding i when it generates artefacts that j considers to be p-creative.

Indirectly, i can also affect the evaluation of p-creativity by j because j's evaluation of p-creativity involves an evaluation of novelty, which is partly or wholly based on artefacts it has previously experienced. Hence, i effects a change in j's evaluation

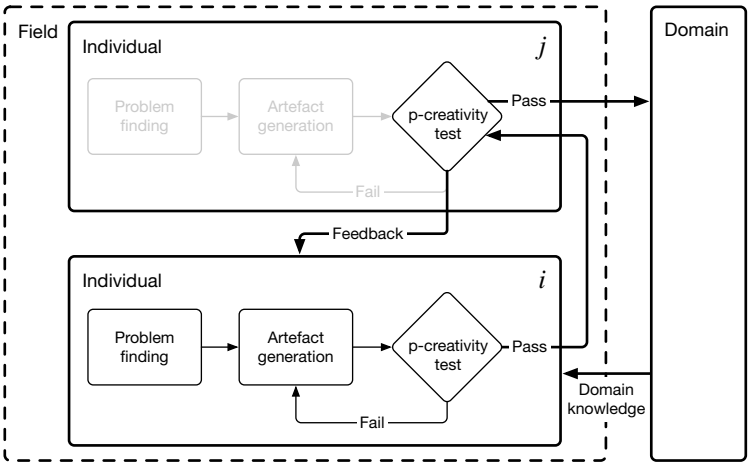


Fig. 14.3 A minimal social creativity test in an artificial creative system. Based on an illustration from Saunders (2002).

of p-creativity every time it causes j to evaluate an artefact and update the prototypes held in memory. By exposing j to artefacts that i considers to be p-creative, it can alter j's evaluation of novelty and hence its p-creativity test.

To implement the socio-cultural creativity test as a collective function of p-creativity tests, a communication policy is required. A simple communication policy, implemented in the system described in Section 14.4, is for agents to communicate an artefact when their evaluation of that artefact's p-creativity is greater than some fixed threshold. In addition, agents have a domain interaction policy; for example, agents may add artefacts, generated by other agents, if the p-creativity evaluation of the artefact is greater than a domain submission threshold. In this way, no individual is allowed to submit their own work to the domain, and thus at least one other agent must find an individual's work creative before it is entered into the domain.

14.3.3 Individuals

An individual in the DIFI framework must be able to transform knowledge from the domain and produce some novelty for the field to determine whether or not it is creative. Consequently, there are three main requirements for an agent-based model of an individual; (1) it must be able to access the contents of the domain, (2) it must be able to generate some novelty and (3) it must be able to communicate with other members of the field.

Given a simple repository-style model of a domain, the ability to access the contents of the domain can be accomplished with a suitable interface for querying the repository, for example, a database. For more complex models of a domain, for

example where some of the knowledge held in the domain may be distributed across a group of agents, then an individual agent may need to communicate with other agents to gain access to their knowledge, as in the case of Gabora's MAV (Gabora, 1995).

In common with other multi-agent-based models, individual agents must be able to communicate with members of the field. Simple message-passing protocols can be used to accomplish this. As a minimum, individual agents can pass artefacts to other members of the field and receive feedback in return. More complex models of communication may include meta-information about artefacts, for example a description using a domain-specific language (Saunders & Grace, 2008).

The ability of an individual to generate some novelty, or, more precisely, the ability to detect that some potentially interesting novelty has been generated, poses the greatest challenge. While the mechanics of producing novelty can be implemented in a variety of ways, the ability to detect novelty and use this to implement a test for 'p-creativity' places specific requirements on the agent. The following describes the components of an agent that uses a novelty detector and a 'hedonic function' to achieve these requirements.

14.3.3.1 Novelty Detection

A novelty detector determines the novelty of a new input based on a model of the expected inputs. Novelty detectors can be implemented in different ways depending on the type of novelty to be detected (Markou & Singh, 2003a, 2003b). One way to implement a novelty detector is to use a classifier that has been trained on a set of expected stimuli, such that when a new stimulus is presented to the classifier, the classification error is an indication of the novelty. In such an agent-based model, an agent i has a memory M_i with K learned categories of artefacts, such that $M_i = \{m_i^1, \ldots, m_i^K\}$, where m_i^k is the kth learned category. Given an artefact a, the novelty detector calculates the novelty, $N_i(a)$, to be

$$N_i(a) = \min_{m_i^k \in M_i} \Delta(a, m_i^k) \tag{14.1}$$

where $\Delta(a, m_i^k)$ is the classification error, which measures the difference between an input a and the kth learned category m_i^k. How the classification error Δ is calculated will differ between implementations; for example, it may be the Euclidean distance from a prototype or a function of the error generated by a neural network.

Simply detecting novelty is sufficient for many applications, for example monitoring of equipment to identify potential faults (Markou & Singh, 2003a), but in computational models of p-creativity more is required. An agent in an artificial creative system needs to be able to identify potentially interesting novelty by modelling a preference for novelty, which can be achieved with the use of a hedonic function.

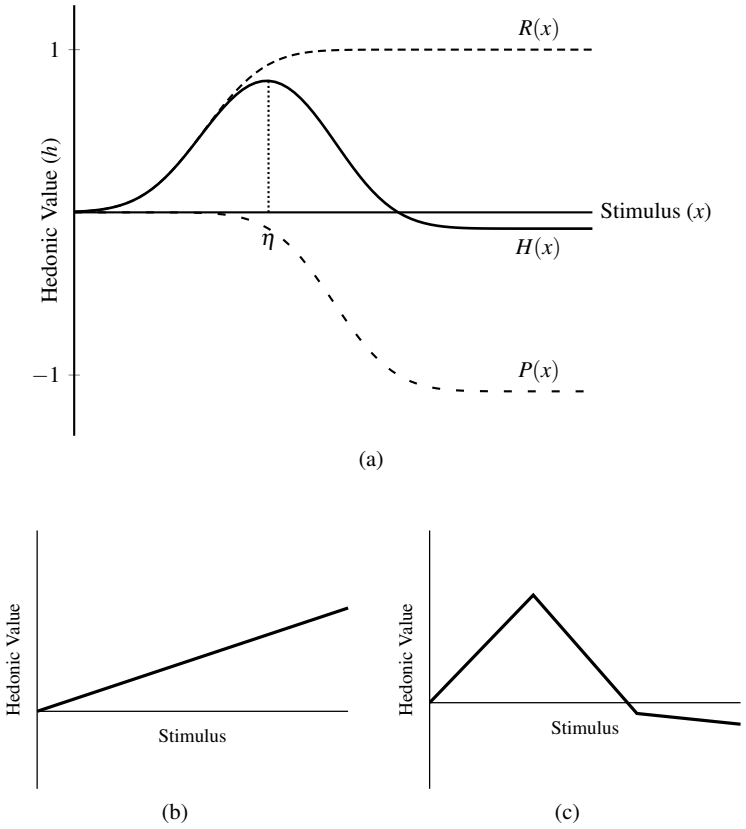


Fig. 14.4 Example hedonic functions (a) Wundt curve, (b) linear and (c) piecewise approximations of the Wundt curve. Illustration of Wundt curve based on Saunders (2002), after Berlyne (1960).

14.3.3.2 Hedonic Functions

A hedonic function defines a transformation from a perceived stimulus to a response signal, which can be used to guide learning and action, for example, in intrinsically-motivated reinforcement learning (Chentanez, Barto, & Singh, 2005). Studies of human preference suggest an inverted U-shape relationship between stimuli and interest (Heckhausen & Heckhausen, 2008; Wundt, 1910), known as the Wundt Curve, illustrated in Fig. 14.4a. For an agent i, the Wundt curve may be implemented as a function, $H_i(x)$, which takes a stimulus, x, and calculates a response signal as the difference of two cumulative Gaussian functions, a reward function $R_i(x)$ and a punishment function $P_i(x)$, which, according to Berlyne (1960), represent a positive

response to small amounts of stimuli and a negative response to large amounts of stimuli:

$$\begin{aligned} H_i(x) &= R_i(x) - \alpha P_i(x) \\ R_i(x) &= F(x \mid \mu_\mathrm{r}, \sigma_\mathrm{r}) \\ P_i(x) &= F(x \mid \mu_\mathrm{p}, \sigma_\mathrm{p}) \end{aligned} \tag{14.2}$$

where μ_r and σ_r are the mean and standard deviation that define the underlying normal distribution for rewarding smaller amounts of stimulus and μ_p and σ_p define the underlying normal distribution for punishing larger stimuli, and α defines the degree to which large values of the stimulus are punished, i.e. for values greater than 1 a negative reward value will be generated for values of x when $P_i(x) \geq R_i(x)$. The cumulative Gaussian function $F(x \mid \mu, \sigma)$ is calculated as

$$F(x \mid \mu, \sigma) = \frac{1}{2}\left[1 + \mathrm{erf}\left(\frac{x-\mu}{\sigma\sqrt{2}}\right)\right] \tag{14.3}$$

where $\mathrm{erf}(y)$ is the Gauss error function, which can be approximated for efficient calculation. It is also useful to define η, as the value of x that generates the peak response:

$$\eta = \arg\max_x H_i(x) = \{x \mid \forall x' : H_i(x') \leq H_i(x)\}. \tag{14.4}$$

Berlyne (1960) identified the Wundt curve as a model for typical reactions that animals and humans display to the presence of novel situations. That is, the most interesting experiences are those that are similar-yet-different to those that have been experienced before, or might be expected given previous experiences. Given a stimulus, $x = N_i(a)$, the Wundt curve can be used to calculate a reward based on novelty. Where x is due to novelty, η represents the preferred novelty for an individual agent. By altering parameters controlling the reward and punishment functions, the value of η can be altered to control how novel an artefact must be for it to be considered 'interesting'.

Other hedonic functions are possible and may be desirable for certain domains. For example, if the expected novelty of any artefact can be reasonably assumed to lie within a range of values close to the origin, then a simple linear response function may be sufficient, such as that illustrated in Fig. 14.4b. Alternatively, if the interest values determined by a hedonic function are used only for comparison, for example to compare the relative interest due to different artefacts, such that the absolute value of the interest is not important, then a piecewise linear approximation to the Wundt curve, Fig. 14.4c, may be a suitable approximation.

14.3.3.3 Interest, Boredom and Curiosity

Given a hedonic function, an agent can determine a measure of interest and determine what action to take as a consequence. For example, given a communication threshold τ_C, an agent i may decide to send an artefact a to another agent if $H_i(N_i(a)) > \tau_C$. Alternatively, given a domain submission threshold τ_D, an agent may decide to submit an artefact a to the domain if $H_i(N_i(a)) > \tau_D$. The rules governing when these decisions may be acted upon form a policy for how the field is structured. A measure of interest for each artefact allows an agent to monitor the frequency with which it encounters 'interesting' artefacts. By keeping an accumulated measure of interest over time, it is possible to develop a simple computational model of 'boredom'. Accumulating interest over time can be achieved simply, for example using $S_i = \alpha S_i + (1 - \alpha)H_i(N_i(a))$, where S_i is the accumulated interest for agent i and α is a suitable decay rate. A state of 'boredom' can then be modelled whenever $S_i < \tau_B$, where τ_B is a suitable threshold for a desired minimum interest level that the agent attempts to maintain.

Given a model of boredom, it is possible to model a type of curiosity identified by Berlyne (1960) as 'diversive curiosity'. In diversive curiosity, a lack of novel stimuli produces a change in behaviour to increase potential exposure to new experiences. An agent in an artificial creative system may implement this type of curiosity very simply by retrieving an artefact from the domain. Alternatively, an agent could adjust the parameters of its generative system such that a more diverse range of artefacts are produced. This simple model of curiosity is based on an assumption that the memory of an agent contains an implicit model of the expectations of future experiences. More sophisticated explicit models of curiosity have been developed and models with explicit expectations have been developed to model surprise (Baranès & Oudeyer, 2009; Merrick & Maher, 2006; Schmidhuber, 1991). Other types of curiosity have also been identified; for example, 'specific curiosity' may also be computationally modelled. In addition, other forms of intrinsic motivation, for example competence, can be computationally modelled (Merrick & Maher, 2006).

This section has described how multiple agents, interacting with other agents and a repository, can be used to model a creative system. The next section provides a concrete example of an implementation of this approach.

14.4 The Digital Clockwork Muse

The *Digital Clockwork Muse* is an implementation of an artificial creative system inspired by the work of Martindale (1990). In *The Clockwork Muse* Martindale presented an investigation into the role that individual novelty-seeking behaviour plays in literature, music, the visual arts and architecture. Martindale concluded that the search for novelty exerts a significant force on the development of styles. The Digital Clockwork Muse is an attempt to computationally model a creative system to

investigate some of the features of creative societies, driven by the search for novelty as described by Martindale.

Algorithm 1: The Digital Clockwork Muse

while $t <$ *total simulation time* **do**
  foreach *agent i in field F* **do**
    update interest h_i for artefact a_i, $h_i = H_i(N_i(a_i))$
    while *message queue* Q_i *is not empty* **do**
      remove artefact a^n from Q_i, sent by agent n
      calculate the hedonic value $h_i^n = H_i(N_i(a^n))$
      update memory M_i to include a^n
      send feedback including h_i^n to sender agent n
      if $h_i^n >$ *domain submission threshold* (τ_D) **then**
        submit artefact a^n to domain D
      end
      if $h_i^n > h_i$ **then**
        adopt received artefact, $a_i \leftarrow a^n$, $h_i \leftarrow h_i^n$
      end
    end
    generate new artefact a_i' from a_i
    calculate the hedonic value $h_i' = H_i(N_i(a_i'))$
    update memory M_i to include a_i'
    if $h_i' >$ *communication threshold* (τ_C) **then**
      select an agent m from F, where $m \neq i$
      send artefact a_i' to agent m
    end
    if $h_i' > h_i$ **then**
      adopt generated artefact, $a_i \leftarrow a_i'$, $h_i \leftarrow h_i'$
    end
    update interest level $S_i = \alpha S_i + (1 - \alpha)h_i$
    if $S_i <$ *boredom threshold* (τ_B) **then**
      retrieve artefact a^d from domain D
      calculate the hedonic value $h_i^d = H_i(N_i(a^d))$
      update memory M_i to include a^d
      if $h_i^d > h_i$ **then**
        adopt retrieved artefact $a_i \leftarrow a^d$, $h_i \leftarrow h_i^d$
      end
    end
  end
end

The operation of the Digital Clockwork Muse is expressed in Algorithm 1. Every agent, i, in a field maintains a current artefact, a_i, with an associated interest value, h_i. At every step in the simulation, each agent implements up to three phases in order to process artefacts (1) an artefact is received from members of the field, (2) an artefact is generated by the individual and (3) an artefact is retrieved from the domain.

In the first phase, each agent i evaluates every artefact a^n in its message queue, shared by another agent n, to calculate a hedonic value h_i^n. Agent i sends an evaluation

of its interest in the artefact, h_i^n, to the sending agent, n. If the agent calculates that its interest in an artefact exceeds the domain submission threshold τ_D, then the agent adds the artefact to the domain, D. If the agent calculates that its interest in an artefact from the queue exceeds its interest in its current artefact, $h_i^n > h_i$, it will adopt the received artefact, $a_i \leftarrow a^n$.

In the second phase, each agent generates a new artefact, a_i', based on its current one, a_i, and evaluates its interest in the generated artefact, h_i'. If the agent's interest in the generated artefact exceeds the communication threshold τ_C, then the agent will choose another agent, m, from the field and send the generated artefacts to it. If the agent's interest in the generated artefact is greater than its interest in the current artefact, $h_i' > h_i$, it will adopt the generated artefact, $a_i \leftarrow a_i'$.

In the third phase, an agent updates its internal state of accumulated interest, S_i, based on the hedonic value of the current artefact. If the level of accumulated interest falls below the boredom threshold, τ_B, then the agent will retrieve an artefact, a^d, from the domain D. If the agent's interest in the retrieved artefact, h_i^d, exceeds the agent's interest in the current artefact, then the agent will adopt the retrieved artefact.

14.4.1 Experiments

Martindale (1990) illustrated the influence of the search for novelty by individuals in a thought experiment, the 'Law of Novelty'. The Law of Novelty forbids the repetition of word or deed and punishes offenders by ostracising them. Martindale argued that the Law of Novelty was merely a magnification of the reality in creative fields. Some of the consequences of the search for novelty are that individuals that do not innovate appropriately will be ignored in the long run and that the complexity of any one style will increase over time to support the increasing need for novelty.

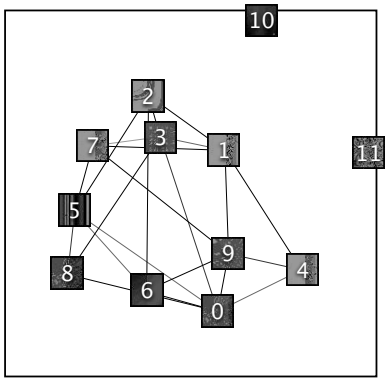
The following experiments were designed to study the effects of the search for novelty in artificial creative societies modelled as agents that have hedonic functions with different preferred novelty values, i.e. η, as defined in Equation (14.4). In this implementation, η ranges from 0 to 32, equal to the range of the potential classification error generated by the novelty detectors used. More detailed accounts of these experiments can be found in Saunders (2002).

14.4.1.1 The Law of Novelty

In the first experiment, a group of 12 agents was created. Ten of the agents, agents 0–9, shared the same hedonic function, i.e. the same preference for novelty ($\eta = 11$). Agent 10 was given a preference for low amounts of novelty ($\eta = 3$) and agent 11 was given a preference for high amounts of novelty ($\eta = 19$). Fig. 14.5b illustrates the network of communication links developed between agents that communicate artefacts and evaluations on a regular basis.

Agent ID	Preferred Novelty(η)	Attributed Creativity
0	11	5.43
1	11	4.49
2	11	4.50
3	11	3.60
4	11	4.48
5	11	1.82
6	11	6.32
7	11	8.93
8	11	10.72
9	11	5.39
10	3	0.00
11	19	0.00

(a)



(b)

Fig. 14.5 The Law of Novelty simulated within a single field of agents with different preferences for novelty (a) attributed creativity between agents; (b) visualisation of the social simulation with two isolated agents.

The results of the simulation are presented in Fig. 14.5a. The results indicate that the agents with the same preference for novelty are somewhat 'creative' according to their peers, with an average attributed creativity of 5.57. Neither agent 10, with a preference for low amounts of novelty, nor agent 11, with a preference for high degrees of novelty, received any credit for their artefacts. Consequently, none of the artefacts produced by these agents were saved in the domain.

The results illustrate the potential for the simulation of the Law of Novelty in artificial creative systems. Agents with a lower novelty preference tend to innovate at a slower rate than those with a higher hedonic preference, and while an agent must produce novelty to be considered creative, it must do so at a pace that matches its audience.

14.4.1.2 The Formation of Cliques

In the second experiment, the behaviour of groups of agents with different hedonic functions was investigated. To do this, a group of 10 agents was created; five of them had a hedonic function that favoured novelty close to $\eta = 6$ and the other five agents favoured novelty values close to $\eta = 15$. Fig. 14.6b illustrates the network of communication of high evaluations between the agents for interesting artefacts.

Two areas of frequent communication can be seen in the matrix of communication shown in Fig. 14.6a. The agents with the same hedonic function frequently send high evaluations of interesting artefacts amongst themselves but rarely send them to agents with a different hedonic function, i.e. there are a large number of high evaluation messages between agents 0–4 and agents 5–9, but only one between the two groups, where agent 4 sends a high evaluation to agent 5.

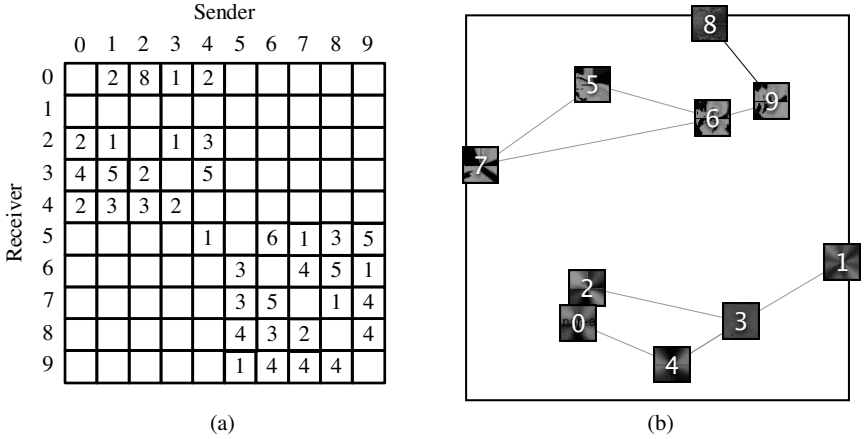


Fig. 14.6 The formation of cliques between agents with different hedonic functions (a) matrix of the number of positive creative evaluations sent between agents, (b) visualisation of social simulation with two non-communicating cliques.

The result of putting collections of agents with different hedonic functions in the same group appears to be the formation of cliques: groups of agents that communicate credit frequently amongst themselves but rarely acknowledge the creativity of agents outside the clique. As a consequence of the lack of communication between the groups, the styles of the artworks produced by the two cliques also remain distinct. The different styles of the two groups can be seen in Fig. 14.6b, with agents 0–4 producing smooth radial images and agents 5–9 producing fractured images with clearly defined edges.

The results of this experiment suggest that when a population of agents contains subgroups with different hedonic functions, the agents in those subgroups form cliques. The agents within a clique communicate high evaluations frequently amongst themselves but rarely to outsiders. The stability of these cliques will depend upon how similar the individuals in different subgroups are and how often the agents in one subgroup are exposed to the artefacts of another subgroup.

Communication between cliques is rare, but it is an important aspect of creative social behaviour. Communication between cliques occurs when two individuals in the different cliques explore design subspaces that are perceptually similar. Each of the individuals is then able to evaluate the other's artefacts highly because they have constructed appropriate perceptual categories. The transfer of artefacts from one clique to another permits new variables in the creative processes of the destination clique; the two cliques can then explore in different directions. Cliques can therefore act as 'super-individuals', exploring a design space as a collective and communicating interesting artefacts within and between cliques.

14.5 Extensions

The agent-based model of social creativity provided by artificial creative systems provides a flexible framework for experimentation, which can be extended in a number of ways to explore different aspects of social creativity. This section explores some examples of these extensions to the domain, the individual, and the field, and interactions between these components.

14.5.1 Domains

The computational model of the domain presented above is lacking in many ways compared with the dynamically evolving source of cultural knowledge that Csikszentmihalyi describes. Saunders (2011) incorporated 'language games' into artificial creative systems to explore the possibility of computationally modelling more complex knowledge structures through the evolution of domain-specific languages. Computational modelling of the evolution of language in creative domains opens up the possibility of investigating computationally a range of important aspects of creativity that are outside the scope of studies focused on individuals, including the emergence of specialised languages that are grounded in the practices of a field, the effects of a common education on the production and evaluation of creative works, and the emergence of subdomains as a consequence of differences in language use across a field.

Wittgenstein (1953) proposed language games as a thought experiment to explore the production of language as a consequence of action and interaction. An example of a language game requires a listener to attempt to identify the topic of an utterance within a given context and for a speaker to provide feedback on the success or failure of the listener. Computational models of language games have demonstrated the ability of agents to evolve languages as a consequence of repeated plays (Steels, 1995). In the extended artificial creative system proposed by Saunders (2011), agents produce utterances to describe artefacts when communicating with other agents. This extended model has been used to explore the impact of the preference for novelty on the formation of creative domains. For the purposes of the computational model, a domain is determined to have formed when a population of agents agree upon a stable lexicon of words with agreed meaning for the associated works. A stable lexicon is said to have formed when communicative success exceeds 80%.

Fig. 14.7 illustrates how individual preference for novelty affects the size of the lexicon and of the ontology of artefacts stored in the domain as a consequence of the field's actions. The results of these simulations show that, for this artificial creative system, increasing the preference for novelty used by individuals to select the topic of a language game has a modest effect on the size of the active lexicon compared with the increase in the size of the active ontology developed across the domain. In other words, the variety of meanings held by a field for a single word increases significantly as a consequence of individuals searching for novel topics. The presence

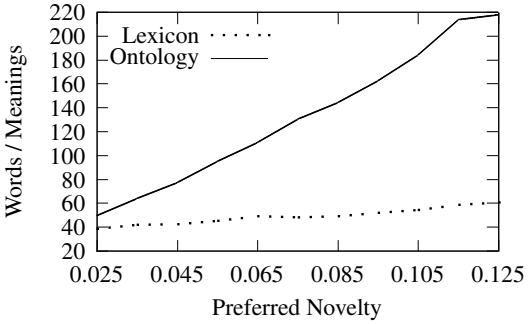


Fig. 14.7 Growth of lexicon and ontology as a consequence of individual preferences for novelty within an artificial creative system.

of ambiguous words in the lexicon of an evolved language has the potential to support computational modelling at the level of domain interactions as a consequence of individuals' actions (Saunders, 2011). This has implications for the modelling of creative processes; the ability to produce and evaluate novel descriptions opens up the possibility of modelling grounded forms of specific curiosity (Berlyne, 1960).

The evolution of domain-specific languages also presents opportunities for domains to differentiate within a single culture, as they present barriers to the flow of information between domains. Consequently, it is possible to computationally model interactions between domains as a result of the actions of individuals (Saunders, 2011).

14.5.2 Individuals

One of the obvious limitations of the computational model of individuals presented here is the lack of an explicit test for the appropriateness of artefacts. Similarly, integrating alternative generative processes, including analogy-making (Falkenhainer, Forbus, & Gentner, 1989), could provide a useful framework for evaluating the effectiveness of such creative processes within a social and cultural context. Curiosity is not the only intrinsic motivation for creative individuals, although it is one of the most persistent (Martindale, 1990). Other motivations for exploring a design space can be modelled computationally, for example competency (Merrick & Maher, 2006).

Building on the development of computational models of intrinsic motivation in robotics (Baranès & Oudeyer, 2009; Marsland, Nehmzow, & Shapiro, 2000), Curious Whispers 2.0 is an example of an embodied artificial creative system, with three robots exchanging simple tunes in much the same way as the software agents in the computational models described above (Saunders et al., 2013). The use of robots

opens up new possibilities of also engaging humans in creative activities. In the case of Curious Whispers 2.0, the robots exchange tunes 'in the open' by performing them and listening for tunes being played by other robots. This openness allowed human participants in the creative system to intervene by playing tunes using a custom synthesiser that could play the same three notes as the robots. The Curious Whispers 2.0 platform has been used to explore interactions between humans and robots when the locus of the creative activity is in the interactions between all of the agents, rather than the human having a privileged role.

14.5.3 Fields

A significant shortcoming of the simulations described above is the small size of the simulated fields. The ability to simulate larger creative societies will permit the study of the spread of innovations and styles. It may also facilitate the emergence of new fields as cliques attain a critical size. Spatial and topological relationships will become more important issues in large population models.

There are several other important players in creative societies besides the producers of innovations (Policastro & Gardner, 1999), including, for example, consumers, distributors and critics. Each has their own role to play in creative societies: consumers evaluate artefacts, distributors distribute artefacts widely and critics distribute their evaluations widely. Convincing other people that you've had a creative idea is often harder than having the idea in the first place.

Building on the extended model of domains described above, Saunders and Grace (2008) introduced 'generation games' as a type of language game where a speaker agent takes the place of a client and an utterance represents a 'brief', such that listener agents, acting as designers, can attempt to satisfy the brief through the production of artefacts. Saunders (2011) also examined the possibility of computational modelling of educators within an artificial creative system.

In non-homogenous societies of agents, the selection of which agents to communicate with becomes an important strategy for agents seeking recognition as a creative individual. Other computational models based on Csikszentmihalyi's system view of creativity have also been developed that demonstrate the important role that authority figures, or 'gatekeepers', play in creative fields (Sosa & Gero, 2005).

14.5.4 Interactions

Artificial societies can have many different policies that control the interactions and decision-making activities of agents. For example, simulations of technological innovation in industry show that consideration of the costs of innovation in decision-making can lead to complex behaviour (Haag & Liedl, 2001). Simulating similar costs in the design process may provide a better understanding of the economics of

creative design in creative societies and the strategies needed to manage creativity with limited resources.

Linkola, Takala, and Toivonen (2016) have implemented artificial creative systems with more complex interactions between the individuals within a field in order to select artefacts that may be added to a domain. In Linkola et al.'s model all artefacts that pass an individual's *self-criticism* test, similar to the p-creativity tests described above, are first published and every agent engages in a two-stage process of voting on which, if any, artefact is to be added to the domain. The first stage of voting allows any agent to veto the addition of an artefact if they assess its novelty to be below a threshold. If any artefacts remain in the set of published artefacts, the second stage of voting selects the one with the highest average novelty assessment from all of the agents to be added to the domain. Linkola et al. explored the effects of varying the self-criticism and veto thresholds on the collective effort required by a field to achieve domains of a given size and concluded that raising the self-criticism threshold reduces the collective effort, while raising the veto threshold maintains the novelty of the artefacts in a domain.

14.6 Conclusion

The computational modelling of creative societies opens up new opportunities for computational creativity that go beyond the modelling of the romantic figure of the lone creative genius or the utilitarian assistant to the human creative. The aim of this chapter has been to present an approach to modelling creativity computationally using multi-agent systems and to show how this can be used to explore aspects of social and cultural creativity. By using agents as models of individuals within creative fields, the framework provides a flexible basis for developing multi-agent systems that can be used to study the interaction between personal and social judgements of creativity. This chapter has also attempted to show that this type of model is open to extension to include other aspects of the social and cultural context for creative individuals, for example domain-specific languages.

There is no doubt that computational modelling will continue to focus on developing analogues for creative cognition and individual creative behaviour. After all, the promise of developing computer programs able to solve problems in ways that are obviously 'creative' is so tantalising that we cannot help ourselves. What this chapter seeks to accomplish, however, is to show that it is possible to develop relatively simple computational models that accommodate both personal and socio-cultural aspects of creativity.

References

Axelrod, R. (1997). The dissemination of culture: A model with local convergence and global polarization. *The Journal of Conflict Resolution*, *41*(2), 203–226.

Baranès, A., & Oudeyer, P.-Y. (2009). R-IAC: Robust intrinsically motivated exploration and active learning. *IEEE Transactions on Autonomous Mental Development*, *1*(3), 155–169.

Berlyne, D. E. (1960). *Conflict, arousal and curiosity*. New York: McGraw-Hill.

Boden, M. A. (1990). *The creative mind: Myths and mechanisms*. London: Cardinal.

Bown, O. (2008). *Theoretical and computational models of cohesion, competition and maladaptation in the evolution of human musical behaviour* (Doctoral dissertation, University of London (Goldsmiths), London).

Bown, O., & Wiggins, G. A. (2005). Modelling musical behaviour in a cultural-evolutionary system. In *Computational creativity workshop*. IJCAI.

Chentanez, N., Barto, A. G., & Singh, S. P. (2005). Intrinsically motivated reinforcement learning. In L. K. Saul, Y. Weiss, & L. Bottou (Eds.), *Advances in neural information processing systems 17* (pp. 1281–1288). MIT Press.

Colton, S. (2012). The painting fool: Stories from building an automated painter. In J. McCormack & M. d'Inverno (Eds.), *Computers and creativity* (pp. 3–38). Berlin: Springer.

Colton, S., Bundy, A., & Walsh, T. (2000). Agent based cooperative theory formation in pure mathematics. In G. Wiggins (Ed.), *Proceedings of AISB 2000 symposium on creative and cultural aspects and applications of ai and cognitive science* (pp. 11–18). Birmingham, UK.

Cope, D. (2005). *Computer models of musical creativity*. Cambridge, MA: MIT Press.

Csikszentmihalyi, M. (1988). Society, culture and person: A systems view of creativity. In R. J. Sternberg (Ed.), *The nature of creativity: Contemporary psychological perspectives* (pp. 325–339). Cambridge, UK: Cambridge University Press.

Csikszentmihalyi, M. (1999). Implications of a systems perspective for the study of creativity. In R. J. Sternberg (Ed.), *Handbook of creativity* (pp. 313–335). Cambridge, UK: Cambridge University Press.

Engeström, Y. (1996). Development as breaking away and opening up: A challenge to Vygotsky and Piaget. *Swiss Journal of Psychology*, *55*, 126–132.

Epstein, J., & Axtell, R. (1996). *Growing artificial societies: Social science from the bottom up*. Brookings Institution Press. Retrieved from https://books.google.com.au/books?id=xXvelSs2caQC

Falkenhainer, B., Forbus, K. D., & Gentner, D. (1989). The structure-mapping engine: Algorithm and examples. *Artificial Intelligence*, *41*, 1–63.

Feldman, D. H., Csikszentmihalyi, M., & Gardner, H. (1994). *Changing the world, a framework for the study of creativity*. Westport, CT: Praeger.

Gabora, L. (1995). Meme and variations: A computer model of cultural evolution. In L. Nadel & D. Stein (Eds.), *1993 lectures in complex systems*. Reading, MA: Addison-Wesley.

Gardner, H. (1993). *Creating minds: An anatomy of creativity seen through the lives of Freud, Einstein, Picasso, Stravinsky, Eliot, Graham and Gandhi*. New York: Basic Books.

Haag, G., & Liedl, P. (2001). Modelling and simulating innovation behaviour within micro-based correlated decision processes. *Journal of Artificial Societies and Social Simulation*, *4*(3). Retrieved from http://jasss.soc.surrey.ac.uk/4/3/3.html

Heckhausen, J., & Heckhausen, H. (2008). *Motivation and action*. New York: Cambridge University Press.

Hoffman, G., & Weinberg, G. (2011). Interactive improvisation with a robotic marimba player. *Autonomous Robots*, *31*(2–3), 133–153. doi:10.1007/s10514-011-9237-0

Lindqvist, G. (2003). Vygotsky's theory of creativity. *Creativity Research Journal*, *15*(2–3), 245–251. doi:10.1080/10400419.2003.9651416. eprint: http://www.tandfonline.com/doi/pdf/10.1080/10400419.2003.9651416

Linkola, S., Takala, T., & Toivonen, H. (2016). Novelty-seeking multi-agent systems. In *Proceedings of the 7th international conference on computational creativity*.

Liu, Y. T. (2000). Creativity or novelty? *Design Studies*, *21*(3), 261–276.

Macedo, L., & Cardoso, A. (2001). Using surprise to create products that get the attention of other agents. In L. Canamero (Ed.), *AAAI fall symposium* (pp. 79–84). Menlo Park, CA: AAAI Press.

Markou, M., & Singh, S. (2003a). Novelty detection: A review–part 1: Statistical approaches. *Signal processing*, *83*(12), 2481–2497.

Markou, M., & Singh, S. (2003b). Novelty detection: A review–part 2: Neural network based approaches. *Signal Process. 83*(12), 2499–2521. doi:10.1016/j.sigpro.2003.07.019

Marsland, S., Nehmzow, U., & Shapiro, J. (2000). Novelty detection for robot neotaxis. In *International symposium on neural computation (NC'2000)* (pp. 554–559).

Martindale, C. (1990). *The clockwork muse*. New York: Basic Books.

Martindale, C., Moore, K., & West, A. (1988). Relationship of preference judgements to typicality, novelty, and mere exposure. *Empirical Studies of the Arts*, *6*(1).

McCorduck, P. (1991). *Aaron's code: Meta-art, Artificial Intelligence, and the work of Harold Cohen*. W.H. Freeman.

Merrick, K., & Maher, M.-L. (2006). Motivated reinforcement learning for non-player characters in persistent computer game worlds. In *ACM SIGCHI international conference on advances in computer entertainment technology, (ACE 2006)*, Los Angeles, CA.

Newell, A., Shaw, J. C., & Simon, H. A. (1962). The process of creative thinking. In H. Gruber, G. Terrell, & M. Wertheimer (Eds.), *Contemporary approaches to creative thinking* (pp. 63–119). New York: Atherton Press.

Policastro, E., & Gardner, H. (1999). From case studies to robust generalizations: An approach to the study of creativity. In R. J. Sternberg (Ed.), *Handbook of creativity* (Chap. 11, pp. 213–225). Cambridge: Cambridge University Press.

Saunders, R. (2002). *Curious design agents and artificial creativity* (Doctoral dissertation, University of Sydney, Australia).

Saunders, R. (2011). Artificial creative systems and the evolution of language. In D. Ventura, P. Gervás, D. F. Harrell, M. L. Maher, A. Pease, & G. Wiggins (Eds.), *Proceedings of the 2nd international conference on computational creativity* (pp. 36–41). Mexico City.

Saunders, R., & Bown, O. (2015). Computational social creativity. *Artificial Life*, *21*(3), 366–378.

Saunders, R., Chee, E., & Gemeinboeck, P. (2013). Human–robot interaction with embodied creative systems. In M. L. Maher, T. Veale, R. Saunders, & O. Bown (Eds.), *Proceedings of the 4th international conference on computational creativity* (pp. 205–209). Sydney, Australia.

Saunders, R., & Gero, J. S. (2001). The Digital Clockwork Muse: A computational model of aesthetic evolution. In G. A. Wiggins (Ed.), *Proceedings of the AISB'01 symposium on AI and creativity in arts and science* (pp. 12–21). SSAISB.

Saunders, R., & Grace, K. (2008). Towards a computational model of creative cultures. In *Proceedings of AAAI spring symposium on creative intelligent systems*, Stanford University, CA.

Sawyer, K. (2012). *Explaining creativity: The science of human innovation* (2nd). Oxford University Press.

Schelling, T. C. (1969). Models of segregation. *American Economic Review*, *59*(2), 488–493.

Schmidhuber, J. (1991). Curious model-building control systems. In *Proceedings of the international joint conference on neural networks* (Vol. 2, pp. 1458–1463). Singapore: IEEE.

Sosa, R., & Gero, J. S. (2004). Diffusion of design ideas: Gatekeeping effects. In H. S. Lee & J. W. Choi (Eds.), *CAADRIA 2004*, Seoul: Yonsei University Press.

Sosa, R., & Gero, J. S. (2005). A computational study of creativity in design: The role of society. *Artificial Intelligence for Engineering Design, Analysis and Manufacturing (AIEDAM)*, *19*(4), 229–244.

Sosa, R., & Gero, J. S. (2008). Creative social systems. In *AAAI spring symposium: Creative intelligent systems* (pp. 90–94).

Steels, L. (1995). A self-organizing spatial vocabulary. *Artificial Life*, *2*(3), 319–332.

Tardif, T. Z., & Sternberg, R. J. (1988). What do we know about creativity? In R. J. Sternberg (Ed.), *The nature of creativity* (pp. 429–440). Cambridge: Cambridge University Press.

Vygotsky, L. S. (1971). *The psychology of art*. Originally published 1930. Cambridge, MA: MIT Press.

Wiggins, G. A. (2008). Computer models of musical creativity: A review of computer models of musical creativity by David Cope. *Literary and Linguistic Computing*, *23*(1), 109–116. doi:10.1093/llc/fqm025. eprint: http://llc.oxfordjournals.org/content/23/1/109.full.pdf+html

Wittgenstein, L. (1953). *Philosophical investigations*. Blackwell.

Wooldridge, M. J. (2001). *An introduction to multiagent systems*. Wiley.

Wundt, W. (1910). *Principles of physiological psychology*. New York: Macmillan.

Chapter 15
Creative Systems: A Biological Perspective

Jon McCormack

Abstract A creative systems approach examines generalised processes of creativity. Looking beyond the anthropomorphic focus of traditional creativity research, creative systems encompass a broad range of generative processes, including the physical, chemical, biological and social. Looking beyond human creativity opens up the idea of creative processes occurring over timescales or dimensions outside normal human experience. This chapter examines the use of creative systems as a way to conceptualise, design and develop new kinds of creative works, particularly those that make use of computers. In particular, it looks at biological processes and illustrates how they can be abstracted, generalised and repurposed as creative algorithms. Traditional evolutionary approaches to computer creativity focus on optimisation, in that they define some criterion that allows the ranking of individuals from a population in terms of their suitability for a particular task. The problem for creative applications is that creativity is rarely thought of as finding a single optimisation. Reconceptualising the exploration of a creative space using an 'ecosystemic' approach, for example, can lead to more creatively rewarding possibilities.

15.1 Creative Systems and Post-anthropocentric Creativity

Anyone entering the world of creativity research for the first time will find themselves confronted with a broad range of definitions and understandings of creativity (see other chapters in this volume, for example). There are many factors that contribute to this dilemma, including the fact that the present concept of creativity is only relatively new and is not universally understood in the same way, particularly outside Western culture (Still & d'Inverno, 2016). Current conceptions of creativity, both human and non-human, have only emerged in relatively recent research.

Jon McCormack
Monash University, Caulfield East, Victoria 3145, Australia
e-mail: Jon.McCormack@monash.edu

T. Veale, F. A. Cardoso (eds.), *Computational Creativity*, Computational Synthesis and Creative Systems, https://doi.org/10.1007/978-3-319-43610-4_15

If we take Boden's popular definition – that creativity involves the generation of ideas or artefacts that are *new*, *surprising*, and *valuable* (Boden, 2010) – then an interesting question to ask is: what are the mechanisms that enable this creativity? It appears likely that any such mechanisms are numerous and diverse. While creativity is commonly associated with the human individual, groups, societies and nature invent, too.

The psychologist David Perkins (1996) spoke of 'creative systems'; recognising that there are different mechanisms or classes of underlying systems that are all capable of producing creative artefacts. A creative system, in this view, is capable of producing adaptive novelty in a given context. This suggests natural selection is a creative system, generating things like prokaryotes, multicellularity, eusociality and language, all through a non-teleological process of hereditary replication and selection. Social interaction is another creative system, having given rise to cultural customs such as shaking hands and a variety of grammatical forms in different human languages. The sociocultural view of creativity has gained increasing popularity over the individualist theories so common in much of the early psychology literature on creativity throughout the latter half of the twentieth century (Sawyer, 2011, Chapter 2).

Understanding systems as 'creative' outside of the normal anthropocentric view opens up new possibilities for what constitutes creativity and creative behaviour, bringing a more general theory to the fore (Roudavski & McCormack, 2016). For example processes occurring over timescales and dimensions beyond everyday human experience do not regularly enter the standard creative lexicon, yet they are undoubtedly creative from this systems perspective. Who has not marvelled at the intricacies of major geological formations, or the formation of planets, solar systems and galaxies. Evolution by natural selection occurs over large timescales: the incremental changes have been so slow that for most of human history the process went unnoticed.

Evolution or evolutionary metaphors are often used in describing creative processes e.g., Aunger (2002), Dawkins (1999), Lumsden (1999), Martindale (1999). George Basalla's *The Evolution of Technology* (Basalla, 1998) detailed a theory of technological evolution, offering an explanation for the creative diversity of human-made artefacts: '*novelty* is an integral part of the made world; and a *selection* process operates to choose novel artifacts for replication and addition to the stock of made things'. Evolution has also played an important role in computer-based and computer-assisted creative systems (Bentley & Corne, 2002), being able to discover, for instance, seemingly counter-intuitive designs that significantly exceed any human designs in performance (Eiben & Smith, 2003; Keane & Brown, 1996, p. 10). Such results illustrate the potential of evolutionary systems to devise unconventional yet useful artefacts that lie outside the capabilities of current anthropocentric creative thinking.

Defining a class of phenomena in formal, systemic terms allows for a transition to the computer. The purpose of this chapter is to look at what kinds of computational processes might qualify as 'creative systems', and how computational systems can be embedded in the world as part of a creative system. Here I draw my inspiration

from nature, in particular evolutionary ecosystems. Biological evolution is readily accepted as a creative system, as it is capable of discovering 'appropriate novelty'. The computer science adaptation of evolution, a field known as *evolutionary computing* (EC), selectively abstracts from the processes of biological evolution to solve problems in search, optimisation and learning (Eiben & Smith, 2003). It is important to emphasise *selectively abstracts* here, as only certain components of the natural evolutionary process are used, and these are necessarily highly abstracted from their physical, chemical and biological origins, for both practical and conceptual reasons. In the case of designing a creative system, the challenge is somewhat different from that of standard EC: understanding how a process that is creative in one domain (biology) can be transformed to be creative in another (e.g. the creation of music or art) requires different selective abstractions.

Generating the adaptive novelty exhibited in creative systems can be conceptualised as a process of exploration through a space of possibilities, searching for regions of high creative reward. Perkins (1996) uses the metaphor of the 'Klondike space' – *Gold is where you find it.* He identified four basic problem types in the creative search of a conceptual space (Fig. 15.1, top): (i) *rarity:* viable solutions are sparsely distributed in a vast space of non-viable possibilities; (ii) *isolation:* places of high creative value in the conceptual space are widely separated and disconnected, making them difficult to find; (iii) *oases:* existing solutions offer an oasis that is hard to leave, even though better solutions might exist elsewhere; and (iv) *plateaus:* many parts of the conceptual space are similar, giving no clues as to how to proceed to areas of greater creative reward.

This classification is similar to some archetypical search and optimisation problems encountered in EC (Fig. 15.1, bottom), where algorithms search for optima in what are often difficult phenotypic spaces (Luke, 2009). For example, 'rarity' corresponds to 'needle in a haystack', 'oasis' to 'deceptive'. Noisy landscapes are particularly problematic, where evolutionary methods typically do no better than random search.

Knowing as much as possible about the structure of the space you are searching is immensely important, as it allows you to search strategically using the most efficient methods. Additionally, being able to restructure the space can make it more intuitive for creative exploration. Hence the design of any creative system should consider the structural design of the creative space carefully. It is also important to emphasise that the search process is really an explorative one. For most creative systems, this space is *vast* (McCormack, 2008b), and there may be many isolated 'Klondike spaces' of rich creative reward. The challenge is to efficiently and effectively find and explore them.

15.1.1 Spaces of Possibility

We should make further distinctions about creative spaces and spaces of possibility. As I have previously discussed (McCormack, 2008b), in many domains there are

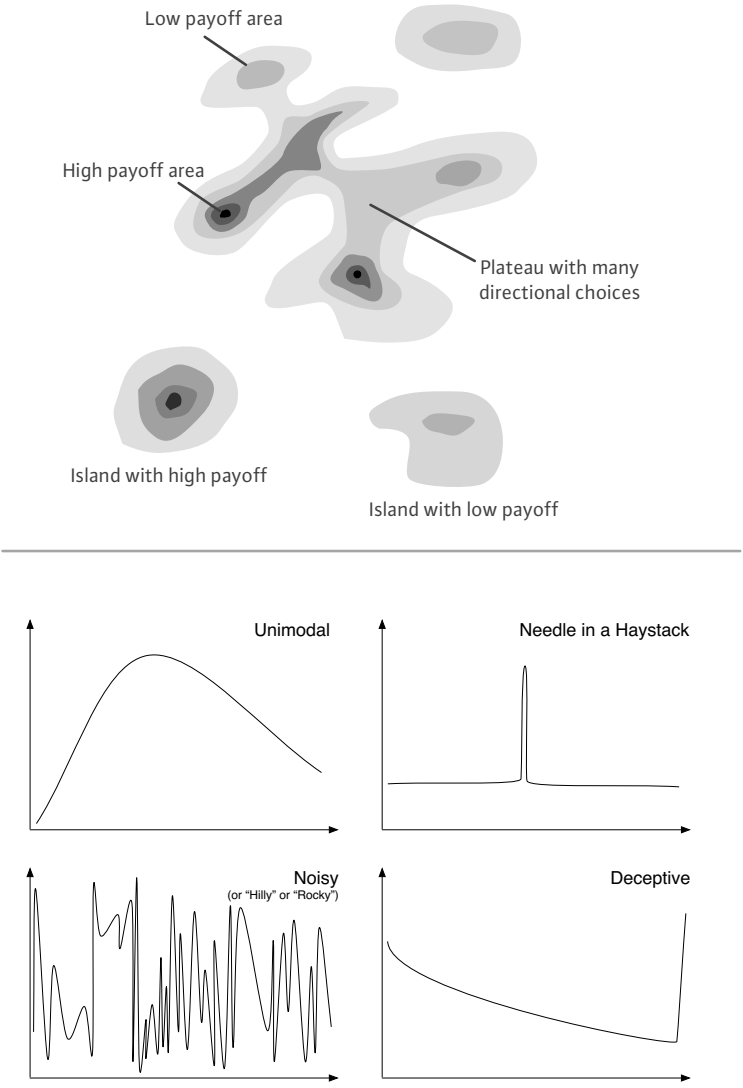


Fig. 15.1 Illustrative diagram of 'Klondike spaces' (top, after Bell (1999)) and characterisation of archetypical search spaces in evolutionary computing (right, after Luke (2009)).

large and crucial differences between the possible and actual. For example, consider a digital image defined by executing an arbitrary Lisp expression over some domain

(x, y), where x and y are the coordinates of a rectangular grid of pixels that comprise the image. Iterating through each coordinate, the expression returns the corresponding pixel's colour. Different expressions will usually generate different images (although many different expressions will also generate the same image). In theory, this system is capable of generating *any* possible image, provided we have the appropriate Lisp expression to generate it.

This represents a space of possibilities that encompasses every possible image that can be represented by coloured pixels over (x, y). For any reasonable image dimensions, the size of this space is vast, far beyond comparisons with astronomical maxima such as the age of the universe or the number of basic subatomic particles estimated to exist in the universe.

However, the *actual* space of images that can be practically created with a Lisp expression is considerably smaller, limited by physical constraints. From the perspective of evolutionary creativity, if we evolve a Lisp expressions using, for example, an interactive genetic algorithm (IGA, see Section 15.2), the actual images produced are all relatively similar and represent an infinitesimally small fraction relative to the possible space of which the system is theoretically capable.[1]

So, while a representational system may theoretically cover a large range of possibilities, searching them – even with evolutionary methods – will only permit examination of insignificantly small regions. Furthermore, transformation or modification of the underlying generative mechanism[2] may open up new spaces not so easily found by the original, for example the addition of symmetry functions in the Lisp expression example would make it easier to generate images with symmetric elements. Of course, we need some way of finding the 'right' transformations or modifications to make to this generative mechanism. This is a kind of 'meta-search' (a search of the different types of generative mechanisms that define a representational space), and in turn evokes an infinite hierarchy (meta-meta-search, meta-meta-meta-search etc.), which effectively amounts to the same problem of the possible and actual in our original 'flat' search.

In practical terms this requires that there must be some human-defined generative mechanism as the basis for any computational creative system,[3] which requires a lot of human ingenuity and creativity if its design is to be effective. I will return to this point in Section 15.4.3. While much research effort and discussion has focused on evaluation and judgement in computational creative systems, representation has received less attention.

A somewhat analogous situation exists in biology. The space of possible DNA sequences is far greater than the space of viable, or possible, phenotypes.[4] The space

[1] By my estimates, about $5 \times 10^{-1444925}\%$ for images of modest dimensions, far beyond astronomically small.

[2] By 'generative mechanism' I am technically referring to the genotype and the mechanism that expresses it into a phenotype.

[3] The mechanism can include the ability to self-modify, change, or learn.

[4] We might think of 'viable' as meaning being able to effectively express a living organism from a zygote or through mitosis of a parent cell. But this is problematic for many reasons, most of which are too tangential to the argument to list here.

of possible phenotypes (those which could exist) is again larger than the space of actual phenotypes (those which have existed or currently exist). In nature, what can be successfully expressed by DNA is limited materially by physical constraints and processes. In contrast to our Lisp expression example, once RNA and DNA were established evolution has not really experimented with different generative mechanisms. We think of DNA as being a highly successful self-replicating molecule, which might be true, but we have little to compare it with. Many factors affect the *variety* of life that has evolved on Earth. As evolution involves successful adaptations, the changing environment of the Earth is an important factor in determining evolutionary variety. In addition to geological events, environments change owing to the presence of species and their interactions, a point that I will return to later in this chapter.

15.2 Evolutionary Computing and Creativity

EC methods (which include techniques such as genetic algorithms, evolutionary strategies and genetic programming) have demonstrated success in assisting users of complex creative systems to better locate regions of high creative reward (Bentley & Corne, 2002; Romero & Machado, 2008). These are 'generate and test' algorithms that evolve a population of candidate solutions or artefacts. New, child artefacts are generated through random mutation and/or recombination with selected parents. Populations are tested or ranked by some measure, with the most highly valued individuals and their offspring more likely to survive in subsequent generations. Incrementally, the overall 'quality' of the population *should* improve according to the fitness measure used. How well the method does depends on many factors, including the nature of the fitness landscape (determined in part by the representational scheme) and the evaluation of the fitness of the solution in artefacts. Success or otherwise is dependent on (i) the structure of the phenotype space, and (ii) the effectiveness of the fitness evaluation in determining the quality of the artefacts produced.

Evolutionary approaches and aesthetic evaluation have been reviewed extensively by Galanter (2012). So it is pertinent here to make just a few points. Firstly, it is important to differentiate between an evolutionary system that gives *creative* results and one that generates *aesthetically pleasing* results. The former does not preclude the latter, but they are, in general, independent (i.e. it is possible for a machine or algorithm to generate aesthetically pleasing images without that system being creative). This distinction is often overlooked, probably owing to a misunderstanding of the multiple meanings of aesthetics in the evolutionary art community (McCormack, 2013).

Some evolutionary systems use learnt or predefined measures of 'creative' features in their generated artefacts (Baluja, Pomerleau, & Jochem, 1994, 2&3; Machado & Cardoso, 2002), or rely on an aesthetic measure to evaluate individual fitness (Birkhoff, 1933; Machado, Romero, & Manaris, 2008; Ramachandran, 2003; Staudek, 2002; Svangård & Nordin, 2004). Others use iterative human selection to rank individuals as part of the evolutionary process (Takagi (2001, 9) provides a comprehensive

survey). These approaches suffer from difficulties, however. Predefined measures of aesthetics, for example, risk implicit judgements as to which specific properties are of value (hence determining *what* will be measured). While a number of researchers describe 'aesthetic universals' of evolutionary origin (Brown, 1991; Dissanayake, 1995; Dutton, 2002; Martindale, 1999; Ramachandran & Hirstein, 1999, 6-7), it has long been proposed that aesthetic values also shift according to individual taste, time and culture. Moreover, aesthetics has many interpretations (Koren, 2010), and in contemporary art surface aesthetic qualities are often downplayed or given little significance in appreciating the creativity of a work. Evolving artefacts exclusively for aesthetic value does not necessarily make them creative.

Some attempts have been made to expressly minimise or remove the aesthetic judgement of a particular individual. This is what is referred to as removing 'the signature' of the artist (Boden, 2010, Chapters 9 and 10). The Drawbots system described by Bird, Husbands, Perris, Bigge, and Brown (2008) attempted to create a line-drawing robot using evolutionary robotics. Researchers defined 'implicit' fitness measures that did not restrict the type of marks the robot drawer should make, including an 'ecological model' that used environmental resource acquisition and expenditure to control drawing. However, the results demonstrated only minimal creativity, and the authors concluded that fitness functions which embodied 'artistic knowledge about "aesthetically pleasing" line patterns' would be necessary if the robot were to make drawings worthy of exhibition to humans.

Using human selection (in an *Interactive Genetic Algorithm,* IGA) suffers from a 'fitness evaluation bottleneck' that reduces the human operator's role to that of a 'pigeon breeder' who quickly fatigues (Dorin, 2001; Takagi, 2001, 9). IGAs are generally more suited to explorations by a non-expert user who is unfamiliar with the generative mechanism being evolved. Here, the IGA allows limited navigation through a space of possibilities without the user necessarily understanding the underlying mechanisms that generate them.[5]

These standard evolutionary approaches, while historically important and capable of significant results, are not able to consistently generate convincingly creative results in many domains. Can we do better? Biology certainly can. A useful insight is in recognising that finding the creative 'Klondike spaces' is not simply an optimisation problem (i.e. finding a global optimum using some fitness criterion). Indeed, for most creative domains the idea of evolving towards a single optimum is counter-intuitive, as an artist or designer normally produces many new artefacts over their professional lifetime. New designs or techniques often 'evolve' from previous ones, offspring of both the originating artist and his or her peers (Basalla, 1998). As Basalla (1998) and others have pointed out using the example of technological evolution, the Western emphasis on individual creativity (reinforced socially through patents and other awards) obscures the important roles played by environmental and social dynamics. Hence:

[5] However, there are exceptions where the IGA has proved useful to expert users as well (e.g. Dahlstedt, 2006; McCormack, 2008a).

The trajectory through a creative space is not one of incrementally optimising towards a single goal or fitness measure, rather it is a complex pathway through a series of intermediate and changing goals, each of which may determine the pathway of the next, and may be creative in its own right.

If we are interested in discovering new creative spaces through the synergetic combination of human intelligence and intuitive structuring and representation of the conceptual space, then there are other possibilities. The evolution of species on earth involves a complex set of interrelated processes and events. For example, species do not exist in isolation from their environment or from other species; together, they form a complex network of interdependencies that may impact on the evolutionary process significantly. What happens if we reconceptualise the search of a creative space using insights from the structure and function of evolutionary biological ecosystems?

15.3 Ecosystems

Ecosystems are a popular yet somewhat nebulous concept that is being increasingly embraced by contemporary culture. Environmental groups want to preserve them, businesses want to successfully strategise and exploit them, and the media are part of them. With sales of Nokia mobile smartphones on the decline, Nokia CEO Stephen Elop bemoaned that fact that his company, unlike its rivals, had failed to create an 'ecosystem': one that encompassed smartphones, the operating system, services and users (Shapshak, 2011). Media theorists speak of 'media ecologies' – the 'dynamic interrelation of processes and objects, beings and things, patterns and matter' (Fuller, 2005). The philosopher Manuel De Landa emphasises the flows of energy and nutrients through ecosystems manifesting themselves as animals and plants, stating that bodies are 'nothing but temporary coagulations in these flows: we capture in our bodies a certain portion of the flow at birth, then release it again when we die and micro-organisms transform us into a new batch of raw materials' (De Landa, 2000).

In the broadest terms, the modern concept of an ecosystem suggests a community of connected but disparate components interacting within an environment. This interaction involves dependency relationships leading to feedback loops of causality. The ecosystem has the ability to self-organise, to dynamically change and adapt in the face of perturbation. It has redundancy and the ability to self-repair. Its mechanisms evoke symbiosis, mutualism and co-dependency, in contrast to pop-cultural interpretations of evolution as exclusively a battle amongst individuals for fitness supremacy. Yet we also speak of 'fragile ecosystems', implying a delicate balance or harmony between elements that can easily be broken by external interference. Any anthropomorphic projection of harmony or stability onto ecosystems is naive, however. The history of evolution is the history of change: species, their diversity, morphology and physical distribution, the chemical composition of the biosphere, the geography of the Earth – all have changed significantly over evolutionary time.

The ecosystem's stability is seemingly transitory, then, tied to the shifts in species distribution and the physiochemical environment.

15.3.1 Biological Ecosystems

Ecosystems and ecology are from the domain of biology, where we find a formal understanding, along with many inspirational ideas on the functional relationships found in real biological ecosystems. Modern ecology is the study of species and their relations to each other and their environment. The term 'ecology' originated with the German biologist and naturalist Ernst Haeckel,[6] who, in 1866, defined it as the 'science of the relationship of the organism to the environment', signifying the importance of different species embedded in specific environments. The term 'ecosystem', from the Greek (οικος, household; σύστημα, union), is attributed to the British ecologist, Sir Arthur Tansley, who refined it from earlier use by fellow botanist Arthur Clapham. It grew out of debates at the time about the similarity of interdependent communities of species to 'complex organisms'. Importantly, Tansley's use of the term 'ecosystem' encompassed 'the inorganic as well as the living components' (Tansley, 1939), recognising that the organism cannot be separated from the environment of the biome, and that ecosystems form 'basic units of nature' (Willis, 1997).

In more modern terms, Scheiner and Willig (2008) nominate seven fundamental principles of ecosystems:

1. Organisms are distributed in space and time in a heterogeneous manner (inclusionary rule).
2. Organisms interact with their abiotic and biotic environments (inclusionary rule).
3. The distributions of organisms and their interactions depend on contingencies (exclusionary rule).
4. Environmental conditions are heterogeneous in space and time (causal rule).
5. Resource are finite and heterogeneous in space and time (causal rule).
6. All organisms are mortal (causal rule).
7. The ecological properties of species are the result of evolution (causal rule).

For those wanting to know more details of the contemporary science, a text such as that by Begon, Townsend, and Harper (2006) provides a useful overview of ecological science.

[6] The Danish biologist Eugen Warming is also credited as the founder of the science of ecology.

15.3.2 Ecosystem Models in the Creative Arts

A vareity of different 'ecosystemic' approaches exist in the arts. Examination finds that they are quite diverse and only loosely drawn from biological concepts, probably owing to multiplicitous and nebulous understandings of ecology outside biology, and various metaphoric interpretations of the ecosystem concept.

15.3.2.1 Design and Architecture

Given the state of human impact on the environment, much theory in landscape and architectural design has sought to bring ideas from ecology and ecosystems into the design lexicon (see e.g. Bell (1999)). Through a greater understanding of nature's processes and functions, it is believed that designers can better integrate human interventions within the landscape, minimising their detritus impact, or at least appreciate how design decisions will effect change to the environment over the life of a project and beyond. In architecture, the field of *design ecologies* seeks connections between biological ecology, human communication, instruction and aesthetics, with an emphasis on 'novel concepts of ecologically informed methodologies of communication through design practice' (Murray, 2011).

Generative design uses processes adopted from evolution as a source of design variation and customisation. It brings a number of desirable features to the design of artefacts, including a means to generate and manage complexity, self-maintenance and self-repair, design novelty and variation (McCormack, Dorin, & Innocent, 2004). As discussed in Section 15.2, evolutionary methods such as IGAs are useful for generative design when the designer has only a rudimentary grasp of the underlying generative mechanism that is being evolved. They permit design changes without the need to understand in detail the configuration or parameter settings that generated the design. The application of generative design to customised manufacture has become feasible in recent years owing to the availability of automated, programmable fabrication devices, such as 3D printers and CNC machines, that can inexpensively translate computer representations into one-off physical objects. This allows physical generative designs to be customised to individual constraints or desires, but on commercial manufacturing scales.

Associating design with ecology and ecological principles might imply the superiority of the natural over human design, and ecosystems creating harmony and stable configurations, 'in tune' with nature and natural surroundings. Ecological processes provide a certain cachet, appeal and authority that conveniently lend both a design and a moral credibility to a project. Such views have been rightly criticised (Kaplinsky, 2006). Evolution needs only to offer adequate solutions – ones that are sufficient for growth, survival and reproduction – not necessarily the best or globally optimal ones. 'Optimality' for evolution is dependent on the environment (obviously, polar bears do not do well in deserts). But it is not that nature has nothing useful to teach us. Moving beyond mimicry, a better understanding of the function and behaviour of real biological ecosystems offers new and rewarding possibilities for

design, along with a greater awareness of how our activities ripple out through the environment and impact on other species.

15.3.2.2 Music and Performance

Waters (2007) uses the concept of a 'performance ecosystem' – one that encompasses composition, performance, performers, instruments and environment. Here music and music making are seen as part of a multilayered, complex dynamical system, operating from the acoustic to the social. Emphasis is placed on the dynamical interactions and, importantly, feedback processes between components of the ecosystem. For example, the feedback between a performer and their instrument encompasses the body, tactility, vibrating materials, and the physical and acoustic properties of the room in which the instrument is played, along with the 'psychological adaptations and adjustments' in the body of the performer, who is deeply connected to, and part of, these interacting elements.

Such connections evoke *autopoiesis* (self-creating and self-maintaining (Maturana & Varela, 1980)) and the cybernetic: instruments can be considered part of a continuum that originates from the body, extending through the instrument and environment. The italian composer, Agostino Di Scipio (2003) seeks a reformulation of what is meant by 'interaction' in a technological performance context and invokes the cybernetic concept of ecosystems and feedback dependencies as a sonic interaction paradigm. This is indicative of a more general sense of failure, in creative contexts, of standard technical approaches to human–computer interaction. These traditional approaches emphasise the functional over the explorative and connected. An alternative view, advocated by Di Scipio and many others, sees interaction as 'a by-product of lower level interdependencies among system components' (Di Scipio, 2003). Components are *adaptive* to their surrounding external conditions and able to *manipulate* them. In the case of sound, this involves a sound ecosystem of sound-generating, sound-listening and sound-modifying components, connected in feedback loops with their acoustic environment. In this configuration, sound itself is the medium in which the ecosystem exists. The coupling of components with their environment allows them to change and reconfigure in response to environmental variation: an environment that the components themselves may be modifying.

15.3.2.3 Visual and Installation Art

My own interactive installation, *Eden* (McCormack, 2001), is a complex artificial ecosystem running in realtime on a two-dimensional lattice of cells, projected into a three-dimensional environment (Fig. 15.2). The virtual simulation in *Eden* includes seasonal variation, a planetary albedo modified by biomass composition (Lenton & Lovelock, 2001), and a simulation of sound propagation and attenuation. Evolving, learning agents modify and adapt to their surroundings. Interestingly, the agents often discover a number of behaviours not explicitly programmed into the system,

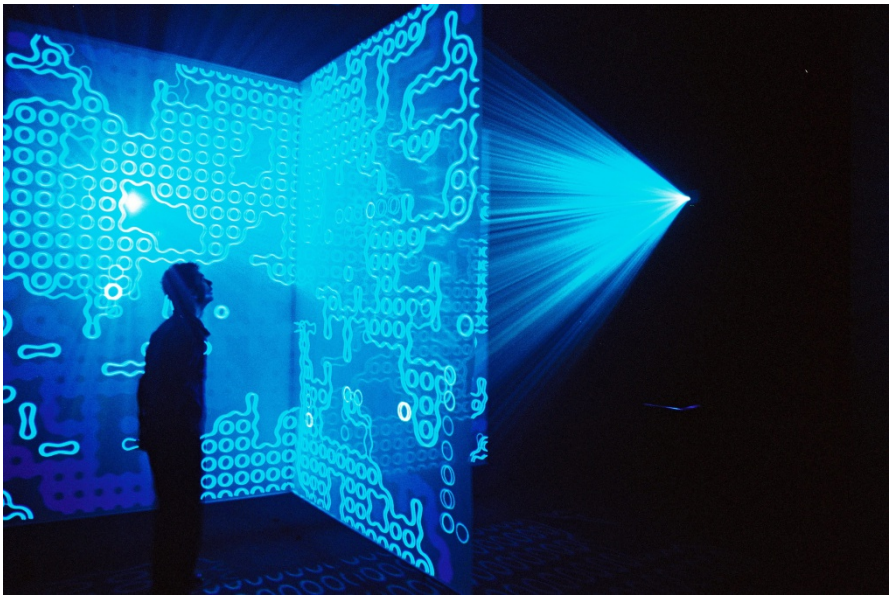


Fig. 15.2 The author's *Eden* installation: an evolving ecosystem of virtual creatures learn new behaviours based on interaction with their environment and with their human audience.

including hibernation during winter months when food resources are scarce, predation and primitive signalling using sound. A computer vision system links human visitor presence to the generation of biomass (food for the agents), and over time agents learn to make interesting sequences of sound in order to keep visitors attracted near the work, thus increasing their supply of food and chances of reproductive success (McCormack, 2009).

Over the last twenty years, the Dutch artists Erwin Driessens and Maria Verstappen[7] have been experimenting with generative 'processes of production' in their art practice. This has extensively encompassed the use of ecosystem metaphors in a number of their works. For example, *E-volver* is a generative visual artwork where a small collection of agents roam a gridded landscape of coloured pixels, choosing to modify the pixel underneath them based on its colour and those of the neighbouring pixels. Each agent has a set of rules that determine how to change the colour and where to move next (Driessens & Verstappen, 2008). Through the interaction of these pixel-modifying agents and their environment (the pixels which comprise the image), *E-volver* is able to generate a fascinating myriad of complex and detailed images (Fig. 15.3 shows one example), all of which begin from a uniformly grey canvas. The images, while abstract, remind the viewer of a landscape viewed from high altitude, an alien mould overwhelming a surface or electron micrographs of some unidentified organic structure. Importantly, they exhibit details on a variety of scales, with coherent structures extending far beyond the one-pixel sensory radius of

[7] See their website at http://www.xs4all.nl/~notnot/index.html

the agents that created them. This suggests a collective self-organisation achieved through agent–environment interaction, with the environment acting as a 'memory' that assists agents in building coherent structures within the image.

Like Di Scipio's sonic ecosystems, *E-volver*'s 'environment' is *the medium itself* (an image comprised of coloured pixels). For *Eden*, the real and virtual environments are causally connected through sound, human presence and the production of resources. In both *E-volver* and *Eden*, agents modify their environment, which, in part, determines their behaviour. Causally coupling agent to environment allows feedback processes to be established, and the system thus becomes self-modifying. This iterative self-modification process facilitates the emergence of heterogeneous order and fractal-like complexity from an environment of relative disorder and simplicity. For *Eden* this is further expanded by the use of an evolutionary learning system (based on a variant of Wilson's XCS (Wilson, 1999)) that introduces new learning behaviours into the system. Learnt behaviours that have been beneficial over an agent's lifetime are passed directly on to their offspring.

Unlike *Eden*'s learning agents, *E-volver*'s agents are not evolutionary over the life of the ecosystem, yet they are evolved: a variation on the IGA allows the user of the system to evolve ecosystem behaviours through aesthetic rejection ('death of the unfittest'). The entire ecosystem – a set of eight agents and their environment – is evolved, not individual agents within a single image. Selection is based on the subjective qualities of the images produced by an individual ecosystem.

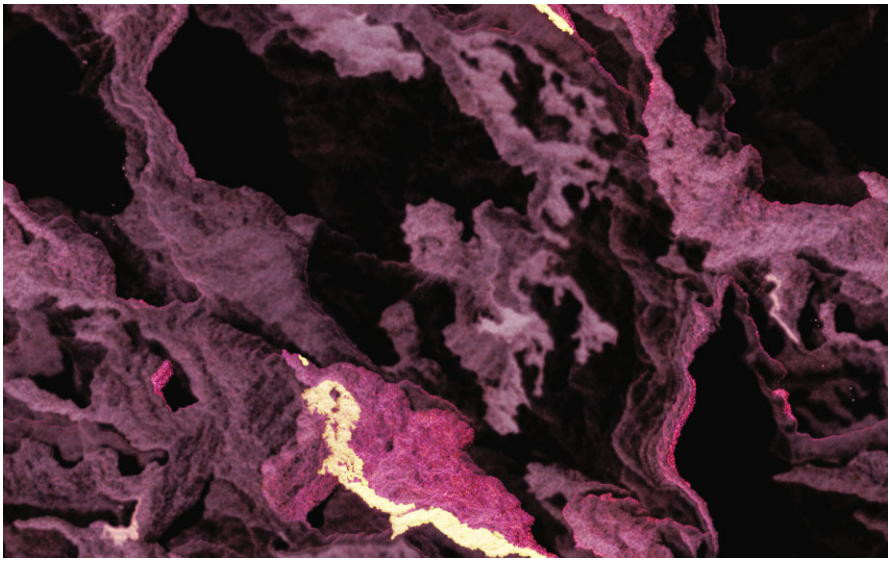


Fig. 15.3 An image produced by Driessens and Verstappen's *E-volver*. Eight pixel-modifying agents build the image by modifying individual pixels. Notice that the image contains coherent structures with multiple levels of detail.

There are numerous other examples of successful artworks based on ecosystem metaphors and processes. To return to the central questions of this chapter: *how* and *why* do they work successfully?

15.4 Ecosystem Design Patterns

Our research group has investigated ecosystemic processes extensively as a basis for designing or enhancing generative artworks (see e.g. Bown & McCormack, 2010; Eldridge & Dorin, 2009; Eldridge, Dorin, & McCormack, 2008; McCormack, 2001, 2007a, 2007b). Our long-term aim has been to develop a catalogue of ecosystemic 'design patterns' in the spirit of Gamma, Helm, Johnson, and Vlissides (1995), which facilitate the building of creative evolutionary systems. Developing these patterns does not imply a 'plug-and-play' approach where one just selects the appropriate patterns, connects them together, and then sits back to watch the creativity evolve. Rather, the patterns serve as starting points in conceptualising a specific creative system, documenting intermediate mechanisms and the typical behaviours they produce. Choosing *which* patterns to use and *how* to apply them remains a matter of significant creative judgement.

Di Scipio (2003) sees the artistic system as a 'gathering of connected components', and it is these components and their interdependencies that must be carefully designed if successful system-level results are to ensue. Components must, additionally, be adaptive to surrounding external conditions and be able to manipulate them.

Table 15.1 summarises the basic properties we think are important to creative ecosystem models. The key to developing a successful ecosystem model is in the design of the system's components and their meaning, interpretation and interaction. In the following sections, I will explore some of these features in more detail, using completed ecosystem artworks as examples of creative systems.

Table 15.1 General properties of creative ecosystem models

Property	Features
Components and their environment	Together these constitute the ecosystem
Dynamical system	Enables the ecosystem to temporally adapt and change in response to internal and external conditions
Self-observation	Provides a link between component action and environment
Self-modification	Allows a component to adjust its behaviour within the system
Interaction	Components must interact with each other and their environment to give rise to emergent behaviours of the system as a whole
Feedback loops	Provide pathways of control, regulation and modification of the ecosystem
Evolution	Allows long-term change, learning and adaptation

15.4.1 Environments: Conditions and Resources

In broad terms, biological environments have two main properties that determine the distribution and abundance of organisms: *conditions* and *resources*. Conditions are physiochemical features of the environment (e.g. temperature, pH, wind speed). An organism's presence may change the conditions of its local environment (e.g. one species of plant may modify local light levels to those which another species is adapted for). Conditions may vary in cyclic patterns or be subject to the uncertainty of prevailing environmental events. Conditions can also serve as stimuli for other organisms. Resources, on the other hand, are consumed by organisms in the course of their growth and reproduction. One organism may become or produce a resource for another through grazing, predation, parasitism or symbiosis, for example.

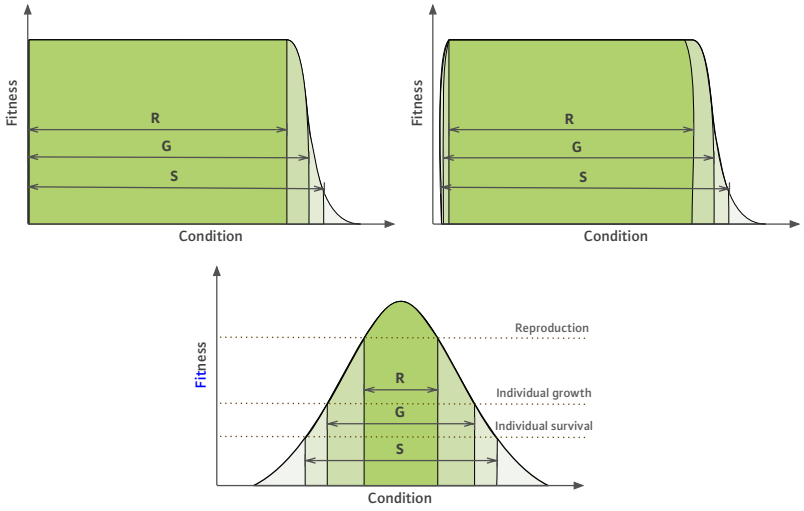


Fig. 15.4 Example organism viability curves for reproduction, growth and survival, from Begon, Townsend, and Harper (2006).

For any particular condition or resource, an organism may have a preferred value or set of values that favour its survival, growth and reproduction. Begon et al. (2006) define three characteristic curves, which show different 'viability zones' for survival, growth and reproduction (Fig. 15.4).

In developing artworks, we can abstract these concepts significantly as long as we are clear about the functional relationships between conditions, resources and organisms. From here on we will consider an organism as a 'component' of an ecosystem; this more genetic term is useful to remind us of the abstractions in

play. Components may often be called ‘agents’ in a computer simulation, typically representing autonomous entities with parameterised, possibly evolving, behaviours.

15.4.2 Self-observation and Feedback

Self-observation gives rise a type of feedback process, similar to a governor or more simply ‘rein control’ (Harvey, 2004). Here ‘observation’ means that the system monitors environmental conditions or resources that are necessary for reproduction, growth and survival and shifts its configuration in response. A component is causally coupled to the environment through relevant conditions or resources within its environment. Observation may be implicit or explicit, local or global. Observation forms a critical connection between a component’s effect on the environment and its ability to modify its behaviour in response, typically to retain homeostasis in local conditions or resources. The use of the term ‘observation’ is deliberately a loaded one. It is used in the cybernetic sense and does not imply a necessary concept of agency (although it does not preclude it). It might be considered the most simple precursor to more complex observational intelligence. It also suggests a system-level (as opposed to an individual-level) ontology that emerges through the interaction of system components.

The well-known model of planetary homeostasis *Daisyworld* uses a simple form of system-level self-observation (Lenton & Lovelock, 2001). Planetary albedo is affected by the proportions of black and white daisies, whose relative proportions change according to surface temperature. What is fascinating about Daisyworld is its ability to maintain a homeostatic surface temperature while the incoming radiant heat energy increases.

In the ecosystem artwork *Colourfield* (McCormack, 2007a), individual components (‘agents’) are bands of colour occupying a one-dimensional lattice of cells. Genetic information controls the colour the agent produces, along with its preference to adapt to the colour of its neighbours and its propensity to occupy vacant neighbouring cells (thus making a larger contribution to the overall colour distribution). A feedback mechanism uses a colour histogram of the overall colour distribution to allocate resources to each individual agent on a per-time-step basis (Fig. 15.5). Here the observation mechanism – resource allocation based on the image histogram – is implicit and global (the system as a whole is observing itself). An individual agent’s contribution to the overall image influences the production of its own resources and those of others. The more cells an individual occupies, the greater the reliance of other individuals on it. Here feedback is an environmental reward function that favours symbiotic adaptations because of its global nature (resources are equally divided between cells). As the system is evolutionary, as a whole it has the ability to modify its colour composition and distribution in response to the ‘self-observation’ provided by this feedback mechanism.

A different self-observation mechanism is in operation in the ecosystem artwork *Niche Constructions* (McCormack, 2010). Niche construction is the process by

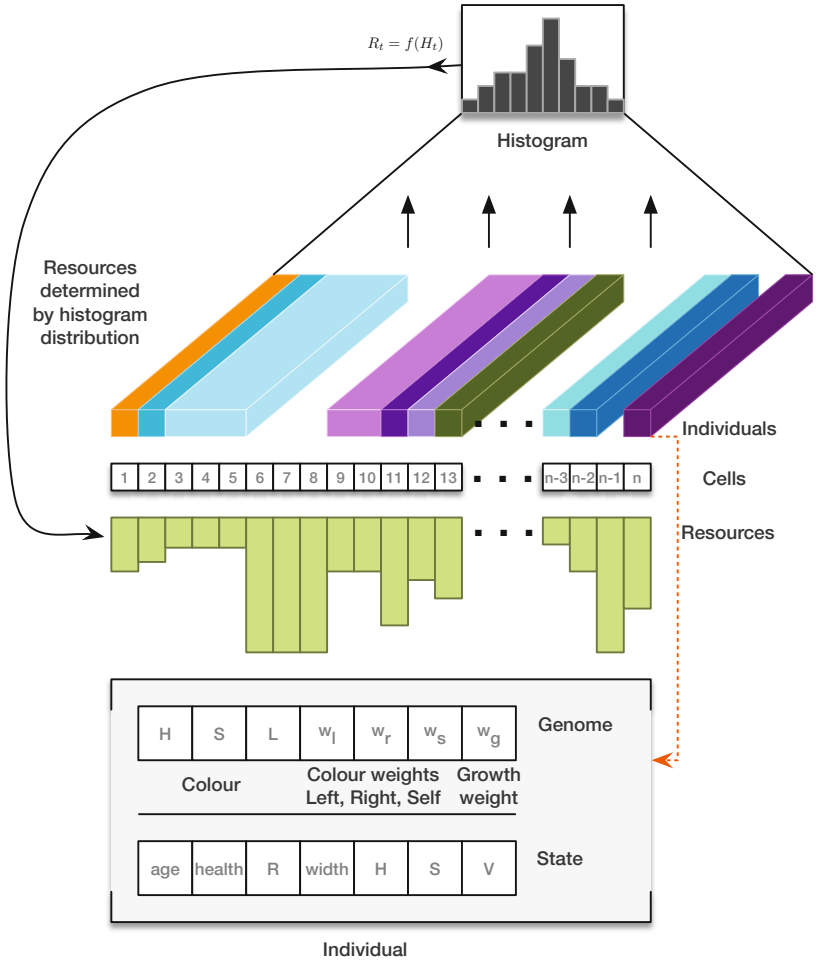


Fig. 15.5 Feedback relationships between components and environment create a form of self-observation in the ecosystemic artwork *Colourfield*.

which organisms, through their activities, modify their heritable environment (and potentially the environments of others). Advocates of niche construction theory in biology argue that it is an initiator of evolutionary change, rather than simply an evolutionary outcome (Odling-Smee, Laland, & Feldman, 2003). The complete set of conditions and resources affecting an organism represent its *niche*, which can be conceptualised as a hypervolume in *n*-dimensional space.

In the *Niche Constructions* artwork, evolutionary line-drawing agents draw on an initially blank canvas as they move around. A set of normalised scalar values

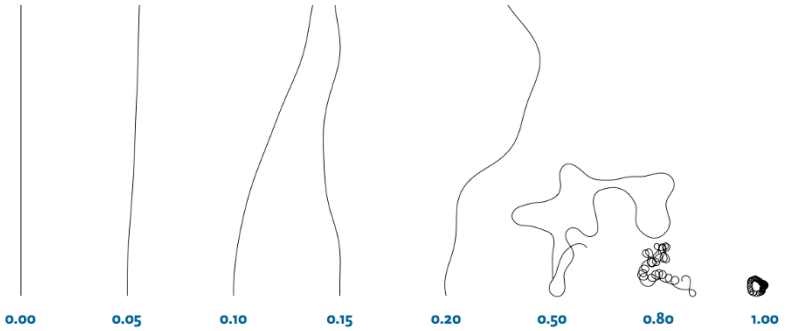


Fig. 15.6 Individual line-drawing agents with different genetic values of irrationality. Note that the 'die if intersect' rule has been turned off for these examples.

forms an agent's genome, which directs its behaviour over its lifetime. Individual alleles control the rate of drawing curvature, 'irrationality' (Fig. 15.6), fecundity and mortality. Agents die if they intersect with any previously drawn line or run off the page. The canvas is seeded with a small initial population of *founder agents* – initialised with uniformly distributed random genomes and positions – that proceed to move, draw and reproduce. There is no limit to the number of offspring an agent may have, but in general the lifespan of agents decreases as the density of lines becomes greater, because it is increasingly difficult to avoid intersection with existing lines. Eventually the entire population dies out and the image is complete. This finished drawing represents the 'fossil record' of all the generations of lines that were able to live over the lifetime of the simulation.

Niche construction is enabled in this work through the addition of a self-observation mechanism that links drawing behaviour genetically to local conditions. As an individual agent draws on the canvas, the local density around it is measured. Each agent has an allele that represents its ideal density preference, i.e. the local line density that is most conducive to its survival, growth and reproduction. As the actual density shifts away from this ideal value, the agent finds it harder to reproduce, grow and survive. If the preferred density and actual density differ too greatly, the agent will die (see Fig. 15.7). Of course, the actual value of this density preference is subject to evolutionary change and, over the life of the drawing, the average density preference increases in the population (McCormack, 2010). The niche construction process influences agent behaviour: low-density-liking agents try to draw large, closed spaces to prevent other lines from decreasing their local density. High-density-seeking agents give birth to large numbers of offspring, which quickly fill the canvas with lines in close proximity. Some examples are shown in Fig. 15.8.

This local, implicit self-observation plays a vital role in influencing the overall density variation and aesthetics of the images produced. We know this because turning the mechanism off produces images of significantly less density variation (statistically) and visual interest (subjectively).

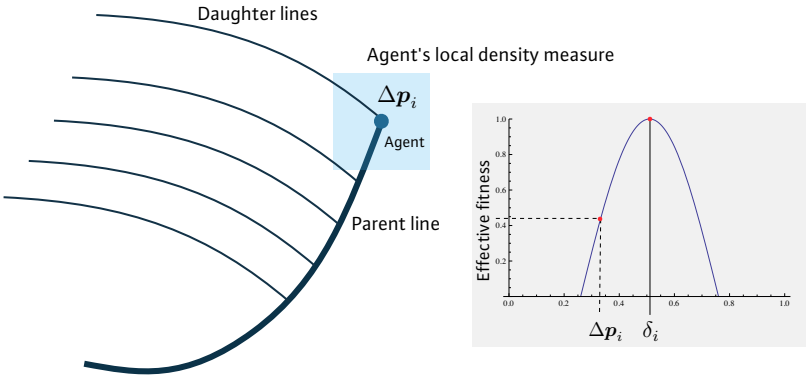


Fig. 15.7 The niche construction mechanism for drawing agents: a local line density measure, Δp_i, facilitates a self-observation mechanism. The agent's genome includes an allele that represents a preferred density, δ_i. The difference between the preferred density and measured density affects the agent's effective fitness, and hence its ability to survive, grow and reproduce.

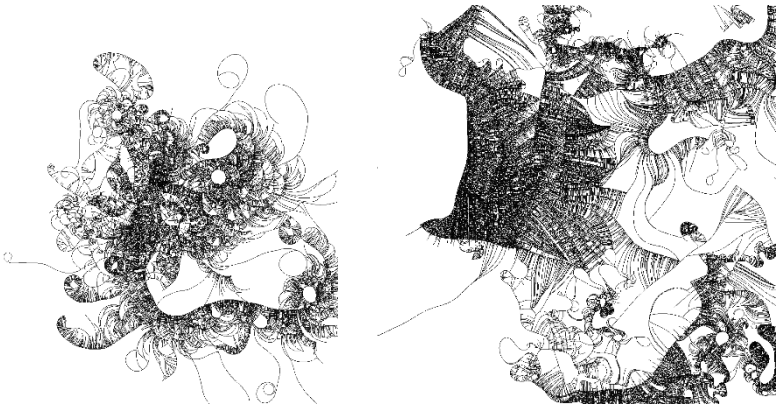


Fig. 15.8 Two sample outputs from the line-drawing system based on niche construction.

A more recent challenge has been to implement this niche construction algorithm in real robots. Mobile, autonomous robots draw on paper with a pen and sense the lines drawn directly below them as they move (Fig. 15.9). An initial version of this system was premiered at the 2015 IJCAI conference in Buenos Aires, Argentina. A significant difference between the robotic and virtual versions is in their evolution. The robots are not able to self-reproduce or make physical copies of themselves, and hence a change in evolutionary dynamics over the lifetime of the drawing is not possible in the current robotic version. Instead, the niche density preference for

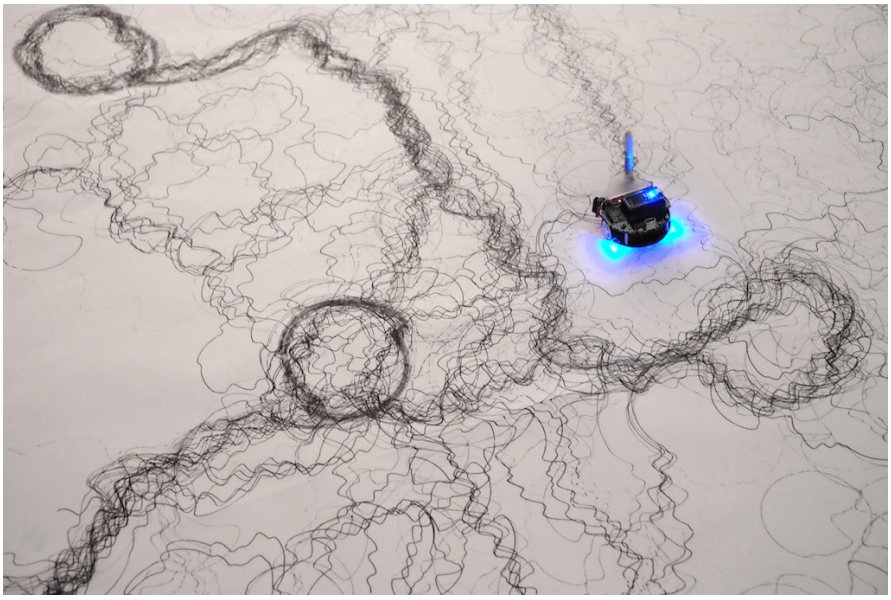


Fig. 15.9 Niche-constructing robot creating a drawing at the ARTE@IJCAI exhibition, Centre Cultural Borges, Buenos Aires, in 2015. The drawings were produced on a 4 m $\times$ 4 m canvas, and took 6–8 hours to complete.

the robot can be adjusted manually via a small potentiometer on the robot. Other parameters for movement are derived from aesthetically pleasing results from evolved virtual line-drawing agents in the earlier software version. Another interesting aspect of the work comes from its physicality – noise and physical parameters of the robot's motion and sensing systems, along with its interaction with the built environment, all affect the drawings produced.

15.4.3 Automation and the Creative Role of the Artist

> automation (noun): the use of largely automatic equipment in a system of manufacturing or other production process
>
> —Oxford Dictionary

The term 'automation' originated in the USA, from the newly industrialised engineering of the 1940s, although similar concepts arose previously in different guises, both historically and geographically. The central idea was to create machines to perform tasks previously performed by humans. The rationale was largely economic: machines that could replace and even outperform their human counterparts would increase production efficiency. As a central driving force in the industrialisation

and technologisation throughout the twentieth century, computers enabled an increasing sophistication and range of capabilities of automation within the capitalist economic system. The idea of machines automating human tasks still underpins many technology-driven approaches to 'automating creativity'. Traditional AI and EC approaches seek the automation of aesthetic or creative optima-finding. In contrast, the ecosystemic approach, as outlined here, does not seek to automate the human out of the creative process, nor claim to equal or better human creative evaluation and judgement. It views creative search and discovery as an *explorative* process, as opposed to an optimisation.

Ecosystemic processes recognise the importance of the link between structure and behaviour. Ecosystem components must be embedded in, and be part of, the medium in which they operate. The design of the system – components and their interdependencies – requires skill and creativity. This design forms the conceptual and aesthetic basis by which the outcomes can be understood. So, rather than removing the artist by automating his or her role, the design ensures that the artist's contribution is one of the utmost creativity – creativity that is enhanced through interaction with the machine. As is also argued elsewhere (McCormack & d'Inverno, 2012), forming an 'ecosystem' that encompasses humans, technology and the socially/technologically mediated environment opens up further ecosystemic possibilities for creative discovery.

There are of course, many reasons why we might seek some form of 'automated creativity' or aesthetic judgement, apart from replacing human labour. For example, automated creativity could lead to creative discovery that exceeds any human capability, or provides greater insights into the mechanisms of human creativity by attempting to model it. But these are 'blue sky' speculations, and current technological advances in this area can just as easily homogenise and suffocate the creative decision-making process for human users as they can expand or enhance it. A good example can be seen in recent digital camera technologies. Over the last ten years, as computational power has escalated, digital cameras have increasingly shifted creative decision making to the camera instead of the person taking the picture. We see modes with labels like 'Intelligent Auto' or scene selection for particular scenarios ('Fireworks','Landscape','Sunset', 'Beach'). These modes supposedly optimise many different parameters to achieve the 'best' shot – all the photographer has to do is frame the image and press the button.[8] Recent advances even take over these decisions, choosing framing by high-level scene analysis and deciding when the picture should be taken based on smile detection, for example. Such functionality trends towards the removal of much human creative decision making, subjugating the human photographer to an increasingly passive role.

As anyone who has used a entirely manual camera knows, hand-operated 'slow technology' forces the user to think about all aspects of the photographic process and their implications for the final image. The user's role is highly active: experimentation, mistakes and serendipitous events are all possible, even encouraged – they are well-known stimuli for creativity. If the *design* of the components and their interaction is good, then using such a device is not marred by complexity or limited by inadequate

[8] This is reminiscent of Kodak founder George Eastman's famous tag line of 1888 for the Kodak No. 1 camera: 'You press the button, we do the rest'.

functionality, which is often the rationalisation given for the automation of creative functionality.

Shifting the thinking about the design of technology from one of 'complexity automation' (where complexity is masked through 'intelligent' simplicity) to one of 'emergent complexity' (where interaction of well-designed components generates new, higher-level functionality) allows the human user to potentially expand their creativity rather than have it subsumed and homogenised.

15.5 Conclusions

Appreciating creativity through a non-anthropocentric, systemic lens has many benefits. Any process that can be formalised into an algorithm can be implemented on a computer, permitting experimentation, analysis and understanding of how the process works and what it can achieve. In effect, the computer becomes a virtual laboratory, where simulations are trialled, hypothesises developed and tested, and theories validated.

A criticism of biologically inspired methods is that they lack intentionality or a 'guiding intelligence' and so cannot be creative. Certainly, evolution by natural selection is a non-teleological process without a designer, without intentionality or any guiding intelligence. Nonetheless, it has been capable of a great many creative innovations: wings, eyes, beaks and brains, to name but a tiny few. The non-anthropocentric view of creativity rejects the idea that creativity requires intention or intelligence, since many non-human creative systems are capable of generating 'appropriate novelty'. What is appropriate to natural selection may not necessarily be appropriate to human concerns, of course. Adapting these processes to human domains, such as design, music or art, requires some forsaking of intention and control, because it is typically difficult or impossible to predict all the possible outcomes. But, as discussed, this is also their advantage: they are not limited by human creativity or human thinking and so are capable of expanding creativity beyond the bounds of conventional human thinking.

Biological processes, such as the ecosystemic methods discussed here, represent an alternative, biologically inspired approach to creative discovery over more traditional methods such as genetic algorithms or genetic programming. They offer an interesting conceptual basis for developing new creative systems and processes, even in non-computational settings or where the computer is only one component of a broader system. Incorporating an 'environment', and allowing interactions between dynamic components and that environment, permits a rich complexity of creative possibilities for the artist wishing to exploit the generative nature of creative systems. While ecosystemic methods do not offer a magic bullet in terms of searching the creative Klondike spaces of any generative system, they do make it easier to at least begin to conceptualise and design systems capable of high creative reward. As the complexity and sophistication of ecosystem artworks develop, we are likely to see further advances in the new creativity made possible with computers that use this

systems approach.

Acknowledgements This research was supported by Australian Research Council Discovery Grants DP0877320, DP1094064 and DP160100166.

References

Aunger, R. (2002). *The electric meme: A new theory of how we think*. New York: Free Press.

Baluja, S., Pomerleau, D., & Jochem, T. (1994). Simulating user's preferences: Towards automated artificial evolution for computer generated images. *Connection Science*, *6*, 325–354.

Basalla, G. (1998). *The evolution of technology*. Cambridge Studies in the History of Science. Matheson Library Main Collection: 609 B297E. Cambridge, UK: Cambridge University Press.

Begon, M., Townsend, C., & Harper, J. (2006). *Ecology: from individuals to ecosystems*. Wiley-Blackwell.

Bell, S. (1999). *Landscape: Pattern, perception and processs*. Hargrave library H712.2 B435L 1999. E & FN Spon.

Bentley, P. J., & Corne, D. W. (Eds.). (2002). *Creative evolutionary systems*. London: Academic Press.

Bird, J., Husbands, P., Perris, M., Bigge, B., & Brown, P. (2008). Implicit fitness functions for evolving a drawing robot. In M. Giacobini, A. Brabazon, S. Cagnoni, G. D. Caro, R. Drechsler, A. Ekárt, ... S. Yang (Eds.), *Applications of Evolutionary Computing, EvoWorkshops 2008* (pp. 473–478). Springer.

Birkhoff, G. D. (1933). *Aesthetic measure*. Cambridge, MA: Harvard University Press.

Boden, M. A. (2010). *Creativity and art: Three roads to surprise*. Oxford University Press.

Bown, O., & McCormack, J. (2010). Taming nature: Tapping the creative potential of ecosystem models in the arts. *Digital Creativity*, *21*(4), 215–231.

Brown, D. E. (1991). *Human universals*. New York: McGraw-Hill.

Dahlstedt, P. (2006). A mutasynth in parameter space: Interactive composition through evolution. *Organised Sound*, *6*(2), 121–124.

Dawkins, R. (1999). *The extended phenotype: The long reach of the gene* (Revised). Oxford; New York: Oxford University Press.

De Landa, M. (2000). *A thousand years of nonlinear history*. Cambridge, Mass: MIT Press.

Di Scipio, A. (2003). 'Sound is the interface': From interactive to ecosystemic signal processing. *Organised Sound*, *8*(3), 269–277.

Dissanayake, E. (1995). *Homo aestheticus: Where art comes from and why*. Seattle: University of Washington Press.

Dorin, A. (2001). Aesthetic fitness and artificial evolution for the selection of imagery from the mythical infinite library. In J. Kelemen & P. Sosík (Eds.), *Advances in artificial life* (Vol. LNAI 2159, pp. 659–668). Prague: Springer-Verlag.

Driessens, E., & Verstappen, M. (2008). Natural processes and artificial procedures. In P. F. Hingston, L. C. Barone, & Z. Michalewicz (Eds.), *Design by evolution: Advances in evolutionary design* (pp. 101–120). Natural Computing Series. Springer.

Dutton, D. (2002). Aesthetic universals. In B. Gaut & D. M. Lopes (Eds.), *The Routledge Companion to Aesthetics*. Routledge. Retrieved from http://www.denisdutton.com/universals.htm

Eiben, A. E., & Smith, J. E. (2003). *Introduction to evolutionary computing*. Natural Computing Series. Springer.

Eldridge, A. C., & Dorin, A. (2009). Filterscape: Energy recycling in a creative ecosystem. In M. Giacobini, A. Brabazon, S. Cagnoni, G. D. Caro, R. Drechsler, A. Ekárt, ... S. Yang (Eds.), *Applications of Evolutionary Computing, EvoWorkshops 2009* (pp. 508–517).

Eldridge, A. C., Dorin, A., & McCormack, J. (2008). Manipulating artificial ecosystems. In M. Giacobini, A. Brabazon, S. Cagnoni, G. D. Caro, R. Drechsler, A. Ekárt, ... S. Yang (Eds.), *Applications of Evolutionary Computing, EvoWorkshops 2008* (pp. 392–401).

Fuller, M. (2005). *Media ecologies: Materialist energies in art and technoculture*. MIT Press.

Galanter, P. (2012). Computational aesthetic evaluation: Past and future. In J. McCormack & M. d'Inverno (Eds.), *Computers and creativity* (Chap. 10, pp. 255–293). doi:10.1007/978-3-642-31727-9

Gamma, E., Helm, R., Johnson, R., & Vlissides, J. M. (1995). *Design patterns: Elements of reusable object-oriented software*. Addison-Wesley professional computing series. Reading, Mass.: Addison-Wesley.

Harvey, I. (2004). Homeostasis and rein control: From daisyworld to active perception. In J. B. Pollack, M. A. Bedau, P. Husbands, T. Ikegami, & R. A. Watson (Eds.), *Ninth International Conference on Artificial Life* (pp. 309–314). MIT Press.

Kaplinsky, J. (2006). Biomimicry versus humanism. *Architectural Design*, *76*(1), 66–71.

Keane, A. J., & Brown, S. M. (1996). The design of a satellite boom with enhanced vibration performance using genetic algorithm techniques. In I. C. Parmee (Ed.), *Conference on adaptive computing in engineering design and control 96* (pp. 107–113). P.E.D.C.

Koren, L. (2010). *Which "Aesthetics" Do You Mean? : Ten Definitions*. Imperfect Publishing.

Lenton, T. M., & Lovelock, J. E. (2001). Daisyworld revisited: Quantifying biological effects on planetary self-regulation. *Tellus*, *53B*(3), 288–305.

Luke, S. (2009). *Essentials of Metaheuristics*. Raleigh, NC: Lulu Publishing.

Lumsden, C. J. (1999). Evolving creative minds: Stories and mechanisms. In R. J. Sternberg (Ed.), *Handbook of Creativity* (Chap. 8, pp. 153–169). Cambridge University Press.

Machado, P., & Cardoso, A. (2002). All the truth about NEvAr. *Applied Intelligence*, *16*(2), 101–118.

Machado, P., Romero, J., & Manaris, B. (2008). Experiments in computational aesthetics. In J. Romero & P. Machado (Eds.), *The art of artificial evolution: A handbook on evolutionary art and music* (pp. 381–415). Natural Computing Series. Springer.

Martindale, C. (1999). Biological bases of creativity. In R. J. Sternberg (Ed.), *Handbook of Creativity* (Chap. 7, pp. 137–152). Cambridge University Press.

Maturana, H. R., & Varela, F. J. (1980). *Autopoiesis and Cognition: the Realization of the Living* (2nd). Dordrecht, Holland: D. Reidel Publishing.

McCormack, J. (2001). Eden: An evolutionary sonic ecosystem. *Advances in Artificial Life, Proceedings of the Sixth European Conference, ECAL*, *LNCS 2159*, 133–142.

McCormack, J. (2007a). Artificial ecosystems for creative discovery. In *Proceedings of the 9th annual conference on genetic and evolutionary computation (GECCO 2007)* (pp. 301–307). ACM.

McCormack, J. (2007b). Creative ecosystems. In A. Cardoso & G. Wiggins (Eds.), *Proceedings of the 4th international joint workshop on computational creativity* (pp. 129–136).

McCormack, J. (2008a). Evolutionary L-systems. In P. F. Hingston, L. C. Barone, & Z. Michalewicz (Eds.), *Design by evolution: Advances in evolutionary design* (pp. 168–196). Natural Computing Series. Springer.

McCormack, J. (2008b). Facing the future: Evolutionary possibilities for human-machine creativity. In P. Machado & J. Romero (Eds.), *The art of artificial evolution: A handbook on evolutionary art and music* (pp. 417–451). Springer.

McCormack, J. (2009). The evolution of sonic ecosystems. In M. Komosinski & A. Adamatzky (Eds.), *Artificial Life Models in Software* (2nd, pp. 393–414). London: Springer.

McCormack, J. (2010). Enhancing creativity with niche construction. In H. Fellerman, M. Dörr, M. M. Hanczyc, L. L. Laursen, S. Maurer, D. Merkle, ... S. Rasmussen (Eds.), *Artificial Life XII* (pp. 525–532). Cambridge, MA: MIT Press.

McCormack, J. (2013). Aesthetics, art, evolution. In P. Machado, J. McDermott, & A. Carballal (Eds.), *Evomusart* (Vol. 7834, pp. 1–12). Lecture Notes in Computer Science. Springer.

McCormack, J., & d'Inverno, M. (Eds.). (2012). *Computers and Creativity*. Springer. Retrieved from http://www.springer.com/us/book/9783642317262

McCormack, J., Dorin, A., & Innocent, T. (2004). Generative design: A paradigm for design research. In J. Redmond, D. Durling, & A. de Bono (Eds.), *Futureground* (Vol. 1: Abstracts, 2: Proceedings, p. 156). Melbourne, Australia: Design Research Society. Retrieved from http://www.designresearchsociety.org/futureground/pdf/689f.pdf

Murray, S. (2011). Design ecologies: Editorial. *Design Ecologies*, *1*(1), 7–9. doi:10.1386/des.1.1.7_2

Odling-Smee, J., Laland, K. N., & Feldman, M. W. (2003). *Niche construction: The neglected process in evolution*. Monographs in Population Biology. Princeton University Press.

Perkins, D. N. (1996). Creativity: Beyond the Darwinian paradigm. In M. Boden (Ed.), *Dimensions of creativity* (Chap. 5, pp. 119–142). MIT Press.

Ramachandran, V. S. (2003). *The emerging mind*. Reith lectures; 2003. London: BBC in association with Profile Books.

Ramachandran, V. S., & Hirstein, W. (1999). The science of art: A neurological theory of aesthetic experience. *Journal of Consciousness Studies*, *6*, 15–51.

Romero, J., & Machado, P. (Eds.). (2008). *The Art of Artificial Evolution: A Handbook on Evolutionary Art and Music*. Natural Computing Series. Springer.

Roudavski, S., & McCormack, J. (2016). Post-anthropcentric creativity. *Digital Creativity*, *27*(1), 3–6.

Sawyer, R. K. (2011). *Explaining creativity: The science of human innovation*. 2nd Edition. Oxford [England]; New York: Oxford University Press.

Scheiner, S. M., & Willig, M. R. (2008). A general theory of ecology. *Theoretical Ecology*, *1*, 21–28.

Shapshak, T. (2011). Why Nokia got into bed with Microsoft. Retrieved from http://www.bizcommunity.africa/Article/410/78/57030.html

Staudek, T. (2002). *Exact aesthetics. object and scene to message* (Doctoral dissertation, Faculty of Informatics, Masaryk University of Brno).

Still, A., & d'Inverno, M. (2016). A history of creativity for future AI research. In F. Pachet, A. Cardoso, V. Corruble, & F. Ghedini (Eds.), *Proceedings of the Seventh International Conference on Computational Creativity (ICCC 2016)* (pp. 147–154).

Svangård, N., & Nordin, P. (2004). Automated aesthetic selection of evolutionary art by distance based classification of genomes and phenomes using the universal similarity metric. In G. R. Raidl, S. Cagnoni, J. Branke, D. Corne, R. Drechsler, Y. Jin, ... G. Squillero (Eds.), *Evoworkshops 2004* (Vol. 3005, pp. 447–456). Lecture Notes in Computer Science. Springer.

Takagi, H. (2001). Interactive evolutionary computation: Fusion of the capabilities of EC optimization and human evaluation. *Proceedings of the IEEE*, *89*, 1275–1296.

Tansley, A. G. (1939). British ecology during the past quarter-century: The plant community and the ecosystem. *Journal of Ecology*, *27*(2), 513–530.

Waters, S. (2007). Performance ecosystems: Ecological approaches to musical interaction. In *EMS07 - The 'languages' of electroacoustic music*, Leicester.

Willis, A. J. (1997). The ecosystem: An evolving concept viewed historically. *Functional Ecology*, *11*(2), 268–271.

Wilson, S. W. (1999). *State of XCS classifier system research*. Prediction Dynamics. Concord, MA: Prediction Dynamics.

Chapter 16
Breaking the Mould
An Evolutionary Quest for Innovation Through Style Change

João Correia, Penousal Machado, Juan Romero, Pedro Martins, and F. Amílcar Cardoso

Abstract An autonomous creative system able to learn, create and innovate is presented. Following previous work on the same topic, the approach explores the interplay between a classifier and an evolutionary system. The classifier is trained with famous paintings and images created by the system, learning to distinguish between these two categories. It is then used to assign fitness, leading to the discovery of imagery that deviates from that previously created by the system. Additionally, by taking phenotype similarity into account, we promote the discovery of diverse images during the course of the evolutionary runs. The images created throughout the evolutionary runs are added to the training set and the process is repeated. This iterative process, which includes retraining the classifier, sets the system into a permanent quest for novelty and innovation. The experimental results obtained across several iterations are presented and analysed, showing the ability of the system to consistently produce novel imagery and to identify atypical images.

João Correia
CISUC, University of Coimbra, Portugal. e-mail: jncor@dei.uc.pt

Penousal Machado
CISUC, University of Coimbra, Portugal. e-mail: machado@dei.uc.pt

Juan Romero
Faculty of Computer Science, University of A Coruña, Spain. e-mail: jj@udc.es

Pedro Martins
CISUC, University of Coimbra, Portugal. E-mail: e-mail: pjmm@dei.uc.pt

F. Amílcar Cardoso
CISUC, University of Coimbra, Portugal. e-mail: amilcar@dei.uc.pt

T. Veale, F. A. Cardoso (eds.), *Computational Creativity*, Computational Synthesis and Creative Systems, https://doi.org/10.1007/978-3-319-43610-4_16

16.1 Introduction

As posited by McCormack (2007), the development of aesthetic judgement systems is one of the biggest challenges in the field of computational creativity. Over the years, two main approaches emerged: the development of hardwired fitness measures that try to encapsulate some sort of aesthetic principle, and the use of machine learning techniques to learn aesthetic models (Romero, Machado, Carballal, & Correia, 2012).

As we have stated in previous works (Machado & Cardoso, 1997; Machado, Romero, & Manaris, 2007; Machado, Romero, Santos, Cardoso, & Pazos, 2007; Romero, Machado, Carballal, & Correia, 2012; Romero, Machado, Santos, & Cardoso, 2003), our long-term goal is the development of artificial artists (AAs) that display the full range of abilities of human artists. In this context, the ability to learn aesthetic models is indispensable, since it gives the system the ability to experience, assess and react not only to its own artistic production, but also to the artworks of other artificial or human artists (Machado & Cardoso, 1997). Furthermore, it also creates the preconditions that allow the system to be inspired by other artists, to detect trends, and to deliberately innovate and deviate.

The ability to consistently generate innovative and adequate artefacts is a key trait of creative human or computational agents. In this chapter, we present an AA that is characterised by the ability to build its own aesthetic model from a set of examples, and by its permanent quest for novelty and innovation through style variation and change.

The approach makes use of an expression-based evolutionary art engine and adaptive classifiers, in this case artificial neural networks (ANNs). The ANNs are trained to discriminate between the artistic production of the system and that of famous artists. The evolutionary engine is used to generate images that the ANNs do not recognise as being products of the system, and that, as such, are novel in relation to its previous artistic practice. During the evolutionary runs, we also promote the discovery of a diverse set of imagery by taking phenotype similarity into account when assigning fitness.

When a set of evolutionary runs is concluded, the novel imagery they have produced is added to the training set, enlarging the area of the search space covered by the system, and the ANNs are retrained. This leads to a refinement of the classifiers, which, in turn, forces the evolutionary algorithm (EA) to explore new paths and styles to break with its past. Thus, the consecutive discovery of new styles is attained through the revision and refinement of the aesthetic criteria for novelty of the AA, while variation within style is attained by promoting phenotype diversity.

The research presented in this chapter builds upon our previous efforts on the same topic (e.g., (Correia, Machado, Romero, & Carballal, 2013b; Machado, Romero, & Manaris, 2007; Machado, Romero, Santos, et al., 2007; Romero, Machado, Carballal, & Correia, 2012)) expanding previous approaches by:

- performing in each iteration of the framework a set of parallel evolutionary runs instead of a single one;

- considering phenotype similarity to promote the discovery of a wide set of diverse images in the course of each evolutionary run;
- using classifiers with access to a larger number of image features;
- using a significantly larger set of training examples;
- using an archive to summarise the innovative imagery produced by the system.

Although we consider that the framework presented here could also be applied to other domains, the scope of this chapter is limited to the domain of images.

The experimental results obtained in several iterations are presented and analysed, showing the ability of the system to consistently produce novel imagery and to identify atypical images without human intervention. We consider that the results obtained in the course of the first iteration are evocative of images produced by user-guided evolution. Furthermore, we claim that the images evolved in the last of the iterations presented are of a significantly different nature, breaking the mould with the previous artistic production of the AA. As such, we hypothesise that a limited form of h- and t-creativity (Boden, 2004) may have been attained.

The chapter is structured as follows. We begin by giving an overview of the state of the art in this field identifying and summarising the work that has been more pertinent to the research presented in this chapter. In Section 3 we give an overview of the EFECTIVE framework, presenting the details of its instantiation in Section 4; this is followed by a presentation and analysis of the experimental results. Finally we draw some conclusions and indicate directions for future research.

16.2 State of the Art

The seminal work of Sims (1991) led to the emergence of a new art form, evolutionary art, which is characterised by the use of evolutionary computation to evolve populations of artworks. In Sims' work, users assign fitness to the images, indicating their favourite ones and, by this means, steering evolution towards regions of the space that match their criteria. This process is in many ways similar to selective breeding, a practice that humans have been following for centuries in the context of animal and plant breeding, to develop, enhance, exaggerate and create particular phenotype traits. This approach to fitness assignment has become known as interactive evolutionary computation (IEC). While IEC has many merits and applications, systems based on IEC are dependent on human users. Therefore, although several computer aided creativity systems have been developed based on IEC (Machado, Romero, Santos, et al., 2007), this approach is not viable for the development of AAs.

Concerning the automation of fitness assignment, the central question is how to develop a scheme that is strongly correlated with human aesthetics or, at least, some aspects of it. One of the most popular approaches to tackling this problem is the use of hardwired fitness functions. There are several notable examples of systems in this category (Greenfield, 2002a, 2003, 2005; Machado & Cardoso, 2002; Machado,

Correia, & Assunção, 2015; Machado, Dias, & Cardoso, 2002; Neufeld, Ross, & Ralph, 2007; Ross, Ralph, & Hai, 2006) and also recent works comparing the merits of such aesthetic measures (Atkins, Klapaukh, Browne, & Zhang, 2010; den Heijer & Eiben, 2010; Ekárt, Joó, Sharma, & Chalakov, 2012; Romero, Machado, Carballal, & Santos, 2012).

The use of machine learning techniques for fitness assignment purposes has also been explored. In their seminal work, Baluja, Pomerlau, and Todd (1994) used an ANN trained with a set of images generated by user-guided evolution to assign fitness. Machado, Romero, and Manaris (2007), Machado, Romero, Santos, et al. (2007) studied the development of AAs able to perform style changes over the course of several runs. In related work, Y. Li, Hu, Chen, and Hu (2012) investigated aesthetic features to model human preferences. The aesthetic model was built by learning both phenotype and genotype features, which were extracted from internal evolutionary images and external real world paintings. Kowaliw, Dorin, and McCormack (2009) compared biomorphs generated randomly, through interactive evolution, and through automatic evolution using a classifier system inspired by content-based image retrieval metrics. The experimental results indicate that the results of the automatic system were comparable to those obtained by interactive evolution. The use of co-evolutionary approaches (Greenfield, 2002b; Saunders, 2001) and hybrid approaches that combine interactive evolution with hardwired fitness functions (Machado, Romero, Cardoso, & Santos, 2005) has also been explored.

Another important contribution of Sims' work concerns the representation. Sims used canonical genetic programming (GP) (Koza, 1992) to evolve images. Here, the genotypes are symbolic expressions, which assume the form of a tree, composed of functions (internal nodes) and terminals (leaves), which may be variables or constants. The phenotypes, i.e. the images, are produced by calculating the outcome of the symbolic expression over a range of values of variables. In other words, the outcome of $expression(x, y)$ yields the colour values of the pixel (x, y) of the image. To produce the entire image, one iterates over the desired range of x and y values, with a given step size.

One of the questions that naturally arises when considering a representation is its expressive power. In this case, what types of images can be represented by means of a symbolic expression of this kind. Although the answer depends, obviously, on the function and terminal set being used, Machado and Cardoso (2002) demonstrated that it is possible to represent any given image using a simple function and terminal set. Provided that the function set contains the `if-then-else` function and that the terminal set contains variables x, y and *constants*, it is trivial to design a symbolic expression for any image, and hence any image is representable. In simple terms, the argument is the following: using `if-then-else`, one can successively partition the image into smaller areas, eventually reaching a pixel level size; then one only needs to use a constant to define the desired pixel colour. Notice that the existence of `if-then-else` is not a strict requirement; as long as there is a way to partition and combine different regions of the image, the idea still holds. Furthermore, many other types of proof are applicable. For instance, if the system has the ability to

encode, explicitly or implicitly, an iterated function system, then one can rely on Barnsley's (1993) proof to demonstrate that all images are representable.

From the above, it is safe to say that most expression-based evolutionary art systems are able to represent any given image. Thus, in theory, it is possible to recreate by evolutionary means any artwork that was ever made or will be made (McCormack, 2007). Practice, however, is an entirely different matter. The images produced by expression-based evolutionary art tend to be abstract and have a mathematical appearance. As pointed out by Romero and Machado (2007), each evolutionary art system tends to have its own signature, which is deeply related to the function set and to the genetic operators being used.

Romero et al. (2003) suggested combining a general purpose evolutionary art system with an image classifier trained to recognise faces, or other types of objects, to evolve images of human faces. In recent years, this idea has been put to practice by several researchers to evolve several kinds of figurative images such as faces, flowers, leaves, breasts, and font glyphs (Correia, Machado, Romero, & Carballal, 2013a; Machado, Correia, & Romero, 2012a, 2012b; Martins, Correia, Costa, & Machado, 2015; Nguyen, Yosinski, & Clune, 2015), as well as ambiguous images (Machado, Vinhas, Correia, & Ekárt, 2015). This kind of approach extended the realm of the imagery produced by expression-based evolutionary art systems, by assigning fitness based on the resemblance to objects that are not usually present in the kind of images these systems tend to produce.

Another approach that has the potential to expand the realm of generated imagery is novelty search. It is important to notice that the use of techniques to promote the novelty of solutions predates the coining of the term novelty search algorithm, by Lehman and Stanley (2008). The work of Saunders (2001) and Romero and Machado (2007) provides examples of early approaches, where novelty plays an important role in evolution, while in the work of Kowaliw et al. (2009), evolution is guided by novelty alone. Among the examples that strictly follow a novelty search mechanism as proposed by Lehman and Stanley, we highlight the work of Secretan et al. (2011) and Liapis, Yannakakis, and Togelius (2013).

Our biggest criticism of canonical novelty search is that we consider that novelty, alone, is not a sufficient criterion for creativity. Furthermore as was analysed in the previous chapter (McCormack, 2019), evolutionary computation approaches have demonstrated success in better locating regions of high creative reward. As such, the work presented in this chapter focuses on three central issues: (i) the automation of fitness assignment; (ii) The development of a system that innovates, overcoming the implicit bias of its representation and expanding the frontiers of its artistic production; and (iii) the generation of artworks that relate to human aesthetics.

16.3 The Framework

The architecture proposed by Romero et al. (2003) argues that an AA should be composed of two main modules: a creator and a critic. The work presented in this

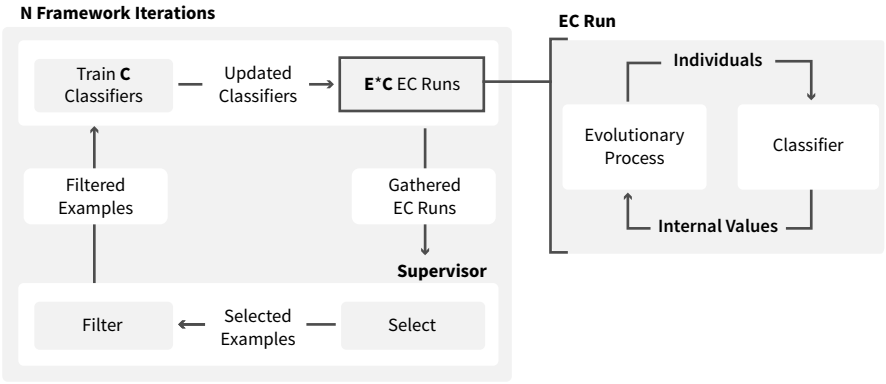


Fig. 16.1 Overview of the EFECTIVE framework.

chapter follows, roughly, this architecture, with the role of the creator being played by an evolutionary computation engine and the role of the critic being played by an ANN. We employ the Evolutionary FramEwork for Classifier assessmenT and ImproVEment (EFECTIVE). In abstract terms, EFECTIVE is a framework that assesses and improves classifier performance through the synthesis of new training instances. In our scenario, it fits the role of the AA, which, based on the its judgement and its past experience, learns iteratively from its inspiration and the work that it has produced, evolving its craft during its existence.

EFECTIVE is composed of three main modules: an EC engine, a classifier system (CS) and a supervisor, each with distinct roles. In brief, the EC engine is responsible for the evolutionary part, where the examples are synthesised and evolved. The CS constitutes the learning approach. The supervisor module manages the examples that result from the interaction between the EC engine and the classifier system, selecting and filtering synthesised examples that will be used to improve the training dataset. These modules come together to create an iterative process for the improvement of classifiers. Figure 16.1 presents an overview of the framework.

Before diving into the details of the instantiation of the framework, we present a succinct description:

1. A set of *external* images is selected. In the case of the work presented in this chapter, this set was composed of famous artworks, representing a source of inspiration for the AA.
2. A set of *internal* images is selected. In the present case, the EC engine is used to randomly create a set of images, thus creating a sample of the type of imagery the evolutionary engine tends to produce.
3. The ANN is trained to distinguish between *internal* and *external* images;
4. A new set of evolutionary runs is started. The output of the ANN is used to assign fitness; Images classified as *external* have higher fitness than those classified as *internal*; Additionally, phenotype diversity is also taken into consideration.

5. During the course of each evolutionary run, an archiving module keeps track of the artistic production of the AA, storing images that are classified as external and diverse from other images classified as external that have evolved during the course of the run.
6. When the set of evolutionary runs is concluded, the supervisor module gathers and merges the archives resulting from each evolutionary run.
7. The consolidated archive is added to the set of *internal* images.
8. The process is repeated from step 3.

One of the key aspects of this approach is the definition of two classes of images. The first class contains *external imagery*. Images that were not created by the GP system and that are usually considered "interesting" or of "high aesthetic value". Conceptually, the external set should be seen as an "inspiration" for the AA. It provides a stable attractor that is meant to ensure that the evolved imagery tends to incorporate aesthetic qualities recognized by humans. The second class contains *internal imagery*, it is composed of images generated by the evolutionary engine, and describes the previous artistic production of the AA. For the purposes of the present work, this class represents undesirable imagery, since we are interested in innovation through style change.

In the present case, the task of the evolutionary module is to evolve images that the ANN classifies as external. This may be accomplished by evolving images that are:

1. Similar to those belonging to the *external* set.
2. Different from the set of internal images (e.g., images that are entirely novel, and hence dissimilar to both sets).

Note that in this context, the concept of similarity and dissimilarity is deeply connected to the features serve as input to the ANN. Therefore, it may deviate from human perception.

The approach relies on promoting competition between the evolutionary engine and the CS. In each iteration, the evolutionary engine must evolve images that are misclassified by the CS, otherwise no progress is achieved. By assigning fitness using a classifier and valuing examples that belong to a predefined class, the approach evolves several misclassified examples (Machado et al., 2012b). These examples are potentially useful for improving the performance of the CS.

The systematic expansion of the internal set, and the subsequent retraining of the ANN, causes an arms race between generator and classifier. As such, from iteration to iteration, the evolutionary engine is forced to explore new paths, which results in stylistic change and in an expansion of the diversity of the artistic production of the system.

As pointed out by Machado, Romero, and Manaris (2007), in the long run, there are two possible final scenarios, which correspond to natural termination criteria for the approach: (i) the evolutionary engine becomes unable to find images that are classified as external; (ii) the ANN becomes unable to discriminate between internal and external imagery. The first outcome reveals a weakness in the evolutionary

engine, which can be caused by a wide variety of factors (deceptive fitness landscape, incorrect parametrisation, lack of computational resources, etc.). In the case of the second outcome, there are two possible subscenarios: (ii.a) the images created by the EC system are similar to some of the external images, which implies that the EC and the CS are performing flawlessly; (ii.b) the images created by the EC system are stylistically different from the external imagery, which indicates a flaw in the CS.

In the next section we present several details pertaining to the instantiation of the framework for the scenario discussed in this chapter. The application of the same framework in other, non-artistic domains has been explored in several publications (Correia et al., 2013a; Machado et al., 2012a, 2012b; Machado, Vinhas, et al., 2015).

16.4 Instantiation of the EFECTIVE Framework

The EFECTIVE framework was instantiated for this scenario with one classifier system, an evolutionary engine and a supervisor. It starts by training a classifier with an initial dataset. Then E parallel evolutionary runs are started. When all evolutionary runs are finished, the supervisor gathers the individuals together and decides which ones are going to be added to the dataset. This cycle is iteratively repeated until a termination criterion is met. The global parameters of the framework are presented in Table 16.1.

Table 16.1 Global parameters of the framework.

Parameter	Setting
Classifiers per iteration (C)	1
EC runs per classifier (E)	30
Adequacy threshold	0.5
Dissimilarity threshold	0.01

16.4.1 Classifier System

The CS is composed of a feature extraction module and an ANN. The participation of the classifier is crucial to the approach for several reasons: it evaluates the images that are generated by the evolutionary engine, and its performance dictates the number of examples that are added and/or deleted before retraining the classifier.

In this work, the CS was trained under certain conditions before it was used to assign fitness during the evolutionary runs. In each training phase the ANN was trained with the full dataset. If the training was entirely successful, meaning that the ANN was able to fully discriminate between the internal and external sets, we proceeded to the evolutionary runs. However, if false externals existed, i.e. if human

produced artworks were classified as evolved images, these images were removed from the external dataset and a new attempt at training was made. Training was only concluded when no external images were classified as internal.

The removal of these images has two motivations. First, from the perspective of the classifier, one can consider that the style these images embody has already been explored. As such, they should no longer be classified as external. Second, from a more pragmatic perspective, these images tend to be atypical in relation to the rest of the images in the external dataset, removing them arguably simplifies the task of the classifier, which may, in turn, result in classifiers that provide fitness landscapes that are more favourable for the evolutionary engine.

The existence of false externals, i.e. evolved images classified as human made, does not have a direct solution. Deleting them would solve nothing. Instead, they remain in the internal dataset. Future iterations are likely to explore the same shortcoming of the classifier, increasing the number of examples of the same style present in the internal dataset, and, owing to the increased cardinality of the subset, forcing the classifier to learn that such images are internal.

16.4.1.1 Feature Extraction

In our approach, the ANNs do not have direct access to the images, instead each image is described by a set of image features, which serve as input to the ANN. So, we developed a feature extractor to extract relevant features from each image.

The pipeline of the feature extractor is the following: the input image is resized to 128×128 pixels and converted to the hue saturation value (HSV) colour space, and a copy of the each image channel is stored for further computation. Several preprocessing operations are computed on demand, depending on the feature to be extracted. A Canny filter is applied to the image and information is extracted from the edges. In total, the feature extraction process yields a total of 120 features, which are later used as input for the classifier.

A thorough description of the feature extractor would be long, and is outside the scope of this chapter. Therefore, we present a brief description of the features extracted, indicating bibliographic references that may provide the interested reader with a complete description. Most of the features implemented originate from previous work concerning the aesthetic analysis of images (Datta, Joshi, Li, & Wang, 2008, 2). The features collected were inspired by the work of Datta, Joshi, Li, and Wang (2006), C. Li and Chen (2009), Faria, Bagley, Ruger, and Breckon (2013), Romero et al. (2003), den Heijer (2012), and based on our own work (Correia et al., 2013b; Machado, Romero, & Manaris, 2007; Machado, Romero, Santos, et al., 2007).

To make the description of the feature set tractable, we introduced a taxonomy (Table 16.2). Some of the features could be classified in several categories, in these cases we followed the consensus in the literature for the feature's category. When selecting and developing this group of features, our goal was to cover several aspects of the images' style and aesthetics.

Table 16.2 The proposed feature taxonomy is composed of five categories: colour, complexity, composition, salience and texture.

Category	Features	Reference
Colour	Palette analysis	
	Average of pixel values	(Datta, Joshi, Li, & Wang, 2006; Machado, Romero, Santos, Cardoso, & Pazos, 2007; Romero, Machado, Santos, & Cardoso, 2003)
	Standard deviation pixel values	(Datta, Joshi, Li, & Wang, 2006; Machado, Romero, Santos, Cardoso, & Pazos, 2007; Romero, Machado, Santos, & Cardoso, 2003)
	Weber contrast	(Faria, Bagley, Ruger, & Breckon, 2013)
	Michelson contrast	(Faria, Bagley, Ruger, & Breckon, 2013)
	Warm and cool colors	(Faria, Bagley, Ruger, & Breckon, 2013)
	Contrasting colors	(Faria, Bagley, Ruger, & Breckon, 2013; C. Li & Chen, 2009)
	Background simplicity	(Datta, Joshi, Li, & Wang, 2006; Rubner, Tomasi, & Guibas, 2000)
Complexity	Fractal dimension	(Machado, Romero, & Manaris, 2007; Romero, Machado, Santos, & Cardoso, 2003)
	Zipf size	(Machado, Romero, & Manaris, 2007; Romero, Machado, Santos, & Cardoso, 2003)
	Zipf rank	(Machado, Romero, & Manaris, 2007; Romero, Machado, Santos, & Cardoso, 2003)
	JPEG and fractal compression	(Correia, Machado, Romero, & Carballal, 2013b; Machado, Romero, & Manaris, 2007; Romero, Machado, Santos, & Cardoso, 2003)
Composition	Horizontal and vertical symmetry	(den Heijer, 2012)
	Liveliness	(den Heijer, 2012)
	Edge density analysis	
	Lighting	(C. Li & Chen, 2009)
	Hue Count	(Datta, Joshi, Li, & Wang, 2006; C. Li & Chen, 2009)
	Blur analysis	(C. Li & Chen, 2009)
	Rule of thirds	(Datta, Joshi, Li, & Wang, 2006)
Salience	Edge distribution	(Datta, Joshi, Li, & Wang, 2006)
	Spatial frequency	(Faria, Bagley, Ruger, & Breckon, 2013)
	Subject size	(Faria, Bagley, Ruger, & Breckon, 2013)
Texture	Tamura contrast	(Tamura, Mori, & Yamawaki, 1978)
	Tamura coarseness	(Tamura, Mori, & Yamawaki, 1978)

Although most of the features are implementations based on the state of the art, we have also introduced some new features in this work. These are briefly described in the following paragraphs.

As the name suggests, the edge density feature captures information regarding the number of edges present in the image. This is achieved by applying a Canny filter to the image and counting the percentage of pixels that correspond to edges, i.e. white pixels.

We have also introduced the palette analysis features, which are intended to provide additional information regarding the image's colour palette. The core idea is to analyse the contrasting colours present in the image (Machado, Correia, & Assunção, 2015). First, we apply a colour quantisation algorithm to reduce the number of colours using k-means clustering. The colour occurrences are counted and sorted in descending order. We compute the distances between the colours of the resulting image using the HSV space, as follows: considering two colour vectors (H,S,V) to represent the colour, the distances between the S and V colour components are calculated using the Euclidean norm; for the H channel, which is circular, we use the formula $dist(a,b) = \min(|a-b|, |a-MAX-b|, |b-MAX-a|)$ to compute the distance, where a and b are two colours and MAX is the maximum value of H. After calculating the distances, we discard the colours that are closer to each other than a predetermined threshold. This results in n colours, which we consider to be the palette of the images. With the palette we calculate a frequency histogram and compute the following metrics: the number of palette colours; the percentage of occurrences; the mode, minimum value and maximum value for each component of the colour; the linear regression and error of the histograms; the average distance to the next colour; the average and standard deviation of the differences between the histogram bins; and the components of the maximum and minimum distance from one colour to the others. We perform the same analysis considering the purity of the colours, which translates into considering only the S and V components of the image's channels to compute the metrics, ignoring the H channel.

16.4.1.2 Artificial Neural Network

The ANN was a feed-forward network, with one hidden layer and two output neurons. It was trained with standard backpropagation. The classifier was built using WEKA's[1] FastNeuralNetwork. WEKA is a workbench for machine learning with a significant number of algorithms and tools available (Hall et al., 2009). The choice of an ANN based classifier is justified by the success of this approach in previous work of related nature (Correia, 2009; Machado, Romero, & Manaris, 2007).

The ANN received the feature vector as input. The output indicates its confidence in classifying the input instance as belonging to the either the internal or the external class. To avoid a binary output, i.e. both neurons returning either 0 or 1, which would result in an unsuitable fitness landscape, we employed a tolerance threshold during

[1] http://www.cs.waikato.ac.nz/ml/weka/, WEKA 3: Data Mining Software in Java

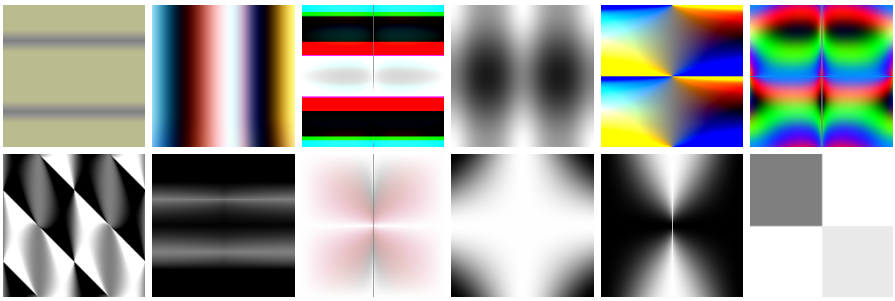


Fig. 16.2 Samples of the internal dataset.

the training stage. This translates into a modification of the training algorithm, where, during the backpropagation of the error, if the difference between the output of the network and the desired output is below the tolerated threshold, then the error is propagated back as zero (no error). The parameters of the ANN are summarised in Table 16.3.

Table 16.3 Parameters related to the ANNs and their training.

Parameter	Setting
Initialisation of weights	Random, interval $[-0.1, 0.1]$
Learning function	Backpropagation
Tolerance threshold	0.3
Learning rate	0.3
Momentum	0.2
Epochs	1000

16.4.2 Initial Datasets

The initial sets of external and internal images play an important role in the performance of our system. We used an external set containing 26238 images including the work of artists such as Cézanne, de Chirico, Dalí, Gauguin, Kandinsky, Klee, Klimt, Matisse, Miró, Modigliani, Monet, Picasso, Renoir and van Gogh. The images were gathered from several different online sources. The rationale was to collect a wide and varied set of artworks. Although we avoided repetitions, it is relatively common for an artist to paint several versions of the same theme. In these cases, and in order to avoid the subjectivity of deciding what was sufficiently different, we decided to include such different variations.

The set of internal images was created using the evolutionary engine, described in the next subsection, to generate 30 initial random populations of size 1600. These

images were added to the internal dataset until the same number of examples existed in the two datasets. Although the images were created randomly, some of the phenotypes could appear more than once. Figure 16.2 presents samples of the images belonging to the internal dataset, illustrating the type of imagery that the EC engine produced in these circumstances.

16.4.3 Evolutionary Engine

For this work we used the ge**N**eral purp**O**se exp**R**ession **B**ased **E**volutionary a**R**t **T**ool (norBErT) as the EC engine (Vinhas, 2015). Inspired by the work of Sims (1991) and Machado and Cardoso (2002), this is a general-purpose, expression-based GP image generation engine that allows the evolution of populations of images. The genotype uses a tree representation to encode individuals and create images from those trees, using a rendering process which consists in generating an output value for each image pixel. Thus, the genotypes are trees composed of a lexicon of functions and terminals. The functions include mathematical and logical operations; the terminal dataset is composed of two variables, x and y, and random constant values and vectors. The phenotypes are images, rendered by evaluating the expression trees for different values of x and y, which serve both as terminal values and as image coordinates. In other words, to determine the value of the pixel coordinates $(0,0)$, one assigns zero to x and y and evaluates the expression tree. In this instantiation the fitness of the individuals is given by the output of the CS, or, more precisely, by the ANN's output as described in Section 16.4.1.2.

As mentioned, we employed a phenotype diversity mechanism by using a novelty search algorithm, designed to evolve a diverse set of adequate images. The main goal of this algorithm is to generate a broader set of images than the set that would be created by a traditional fitness-based EA. In essence, it is a method capable of evolving images according to two criteria that are chosen automatically by analysing the quality of the images produced in each generation. One criterion is to look for the best images according to a fitness function, and the other consists in taking novelty and fitness as two different objectives to be maximised. The reason why novelty is not considered alone is because prior tests have shown how big the search space is and, consequently, how difficult is to discover suitable images (Vinhas, 2015). Similar behaviour has occurred when using a single criterion or considering both fitness and novelty (Vinhas, 2015).

The algorithm's flowchart is similar to that of the traditional EA differing only in two main aspects: (i) the creation of an archive to store the most novel solutions, and (ii) a customised selection mechanism, which is able to consider single or multiple objectives using a tournament-based strategy. The algorithm's flow is shown in Figure 16.3, and can be summarised as follows:

1. Randomly initialise the population.
2. Render the images (phenotypes) from the individuals' genotypes.

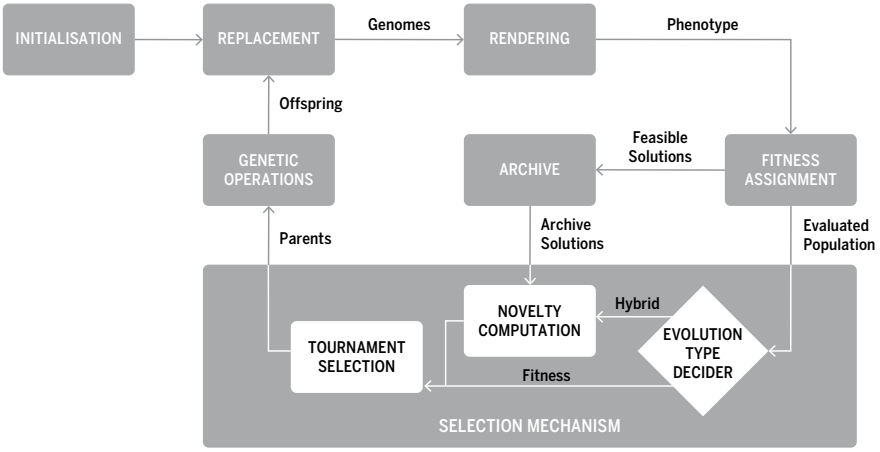


Fig. 16.3 Flow of the proposed hybrid algorithm.

3. Apply the fitness function to the individuals.
4. Select the individuals that meet the criteria to be in the archive (archive assessment).
5. Select the individuals to be used in the breeding process. The individuals are picked using one of the following criteria: (i) according to their fitness, as in a standard EA; and (ii) taking into account both the fitness and the novelty metric, which is computed using the archive members.
6. Employ genetic operators to create a new generation of solutions, which will replace the old one.
7. Repeat the process starting from step 2, until a stop criterion is met.

16.4.3.1 Archive Assessment

In this work, the archive has an unlimited size and plays an important role, because it is used to evaluate our solution and prevents the algorithm from exploring areas of the search space already visited. The idea is that the archive should represent the spectrum of images found to date, and for this reason, the bigger the archive is, the more the algorithm is able to generate suitable and diverse images. Whereas in the work previously mentioned the archive size limited, we opted not to restrict it.

At this stage, a candidate individual has its fitness assigned and it has to meet two requirements in order to be added to the archive: (i) its fitness must be greater or equal than an adequacy threshold f_{min}; and (ii) it needs to be different from those that already belong to the archive. This process is performed by computing the average dissimilarity between the candidate and a set of k nearest neighbours. When the average dissimilarity is above a predefined dissimilarity threshold, $\text{dissim}_{\text{min}}$, the

individual is added to the archive. The values for $f_{\min}$ and $\text{dissim}_{\min}$ are presented in Table 16.4.

The dissimilarity metric for an image i is computed as

$$\text{dissim}(i) = \frac{1}{\text{max}_{\text{arch}}} \sum_{j=1}^{\text{max}_{\text{arch}}} d(i,j), \tag{16.1}$$

where max_{arch} is a predefined parameter which represents the number of most similar images to consider when comparing them with image i, and $d(i,j)$ is a distance metric that measures how different two images (i and j) are. In this dissimilarity measure, however, there are two exceptions that should be highlighted. If there are no entries in the archive, the first individual that has a fitness above $f_{\min}$ is added. Moreover, if the number of archive entries is below max_{arch}, Equation (16.1) is used with the number of archive entries instead of max_{arch}.

For archive assessment, we used to an image similarity metric. Similarity metrics provide us with a notion of distance between pair of images. The development of image distance metrics is an relevant and rich area of research with several applications. A revision of the state of the art is beyond the scope of this chapter. For the interested reader, we suggest the consulting the publications by Wang, Zhang, and Feng (2005) and Goshtasby (2012). Images distance metrics typically involve pixel based operations that can be more or less elaborated. Among the available state of the art options, we chose to employ the normalised cross correlation (NCC), which can be calculated, for two images X and Y with a size of m by n, in the following way:

$$\text{NCC}(X,Y) = \frac{\sum_{i=1}^{m \times n} X_i Y_i}{\sqrt{\sum_{i=1}^{m \times n} X_i^2 \sum_{i=1}^{m \times n} Y_i^2}}, \tag{16.2}$$

where X_i and Y_i correspond to the pixels of images X and Y, respectively.

The NCC similarity outputs a value in the interval $[0,1]$, where 1 indicates the best match. This measure, besides providing a fast calculation, is deemed more robust than most metrics for noisy scenes (Nakhmani & Tannenbaum, 2013). It suits our needs, in the sense that our approach involves a considerable quantity of images, and it can minimise the impact of noisy images on our dissimilarity assessment. So, we used $d(i,j) = 1 - \text{NCC}(i,j)$ as a distance metric.

16.4.3.2 Selection Mechanism

The selection mechanism is important to shaping how evolution will proceed, given the results obtained in a given generation. Our novelty approach includes a customised selection mechanism that can switch between a fitness-based strategy and a hybrid mechanism that considers both fitness and novelty. It starts as a fitness guided evolution; however, that can change according to a decision rule, which can be described as

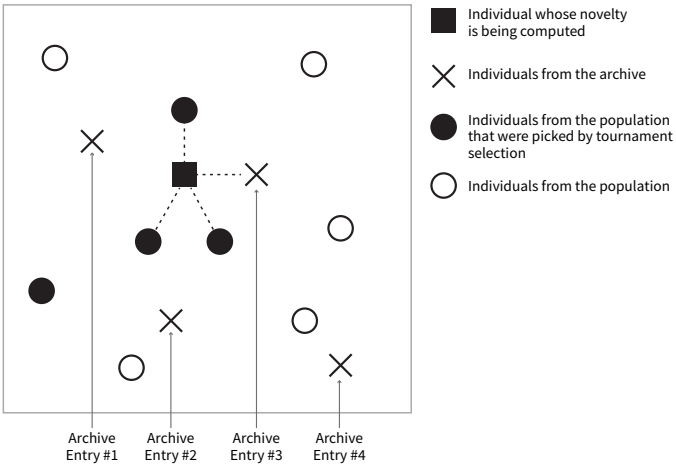


Fig. 16.4 Novelty computation for an individual.

$$\begin{cases} \text{change\_to\_fitness}, & \text{adequate}_{\text{inds}} < T_{\min} \\ \text{change\_to\_hybrid}, & \text{adequate}_{\text{inds}} > T_{\max}, \end{cases}$$

where $\text{adequate}_{\text{inds}}$ is the number of individuals of the current generation that have a fitness above the threshold $f_{\min}$; $T_{\min}$ is the threshold used to verify if evolution should be changed to the fitness-based strategy and $T_{\max}$ is used to decide if it should be changed to the hybrid mechanism.

In fitness-guided evolution, the tournament selection is based on the fitness values of the candidate solutions, as in a standard EA. If hybrid evolution is chosen, it is necessary to compute the novelty of each selected individual and perform a Pareto-based tournament selection, using the novelty and fitness of each selected individual as two different objectives to be maximised.

The novelty computation process is inspired by the work of Lehman and Stanley (2008), with one small change: the k most similar images are taken from the set of the selected individuals and the archive, instead of considering the whole population and the archive. An example of this novelty computation is illustrated in Figure 16.4: for $k = 4$ and a tournament size of 5, the dashed lines denote the individuals chosen to compute the novelty, and it is possible to see that of the four nearest individuals picked, three were chosen from the tournament while the remaining one was chosen from the archive.

At this stage, each selected individual has a fitness and a novelty value, and there is a need to determine the winner of the tournament. This process is inspired by multi-objective EAs, specifically the Pareto-based approaches, which select the best individuals based on their dominance or non-dominance when compared with other individuals. In the present work, the hybrid tournament selection determines the non-dominant solutions by comparing the selected individuals on the basis of both

fitness and novelty. After computing the set of non-dominant individuals, we have the so-called Pareto front. The tournament winner is selected by randomly retrieving one of the solutions on the Pareto front.

The settings chosen for the GP engine and the archive assessment for each EC run are presented in Table 16.4.

Table 16.4 Parameters of the GP engine.

Parameter	Setting
Population size	100
Number of generations	50
Crossover probability	0.8 (per individual)
Mutation probability	0.05 (per node)
Mutation operators	Sub tree swap, sub tree replacement, node insertion, node deletion and mutation
Initialisation method	Ramped half-and-half
Initial maximum depth	5
Mutation max tree depth	3
Archive assessment width	32 px
Archive assessment height	32 px
$T_{\min}$	5
$T_{\max}$	15
Function set	$+$, $-$, $\times$, $/$, min, max, abs, neg, warp, sign, sqrt, pow, mdist, sin, cos, if
Terminal set	x, y, random constants

16.5 Experimental Results

In this section, we present the experimental results obtained using our approach. As previously stated, one of the key characteristics of our approach is its iterative nature. In each iteration we performed 30 evolutionary runs, and once these runs ended, the external images produced by the system, i.e. the images that expanded the range of the artistic production of the system, were added to the internal set and the ANN retrained, promoting the discovery of novel images in subsequent iterations.

We are therefore primarily interested in analysing the differences, in terms of the imagery produced that occur from iteration to iteration. It is impossible to show all the images produced in the course of the evolutionary runs. Even if we were to present only the images classified as external, this would imply presenting $38,283$ images for the first iteration alone. Hence, we will present a synthesis of the results, with which we aim to convey the key experimental findings. We have divided our analysis into subsections as follows: first, we present and examine the results concerning the

Table 16.5 Statistics regarding evolutionary process across iterations. The results pertain to 30 independent evolutionary runs for each iteration of the framework.

Framework iteration	Evolved external images	Images added	Seeds with ev. external	Avg. generations for ev. external
1	38283	30110	30	3.63
2	1426	250	22	18.22
3	816	195	17	20.52
4	752	178	24	25.33
5	366	57	10	29.3
6	1105	433	21	20.24
7	620	131	22	31.64
8	191	31	7	23.86
9	422	115	21	24.90
10	692	62	20	28.5
11	374	126	22	32.27
12	267	101	11	31.09
13	842	352	17	21.76

evolution of fitness over the iterations; next, we inspect the images produced; and finally, we analyse the classifier's training and performance in each iteration.

Although we present results concerning 13 iterations of the framework, it is important to stress that further iterations are still being performed. Therefore, the process is not concluded, and all evidence indicates that a significantly higher number of iterations would be necessary before a *breakdown* of the EC engine or classifier took place.

16.5.1 Analysis of the Numeric Results Concerning Evolution

Table 16.5 depicts a series of statistics concerning the evolutionary process across the iterations, namely the total number of images evolved in the course of the 30 evolutionary runs of each iteration that were classified as external (Evolved external images'); the number of these that were added to the internal set used to train the classifier guiding the next iteration after supervision (Images added); the number of seeds for which the EC engine was able to find at least one image classified as external (Seeds with ev. external); and the average number of generations necessary for finding an image classified as external (Avg. generations for ev. external). This average was calculated taking into account only the seeds where at least one image classified as external was found.

As can be observed, a striking number of images classified as external were found in the course of the first iteration, 38,283, which corresponds to an average of 1276.1 per evolutionary run. All of the evolutionary runs were able to find external images and, on average, they took 3.63 generations to find the first image classified as external. Although this number is somewhat surprising, it is far from being

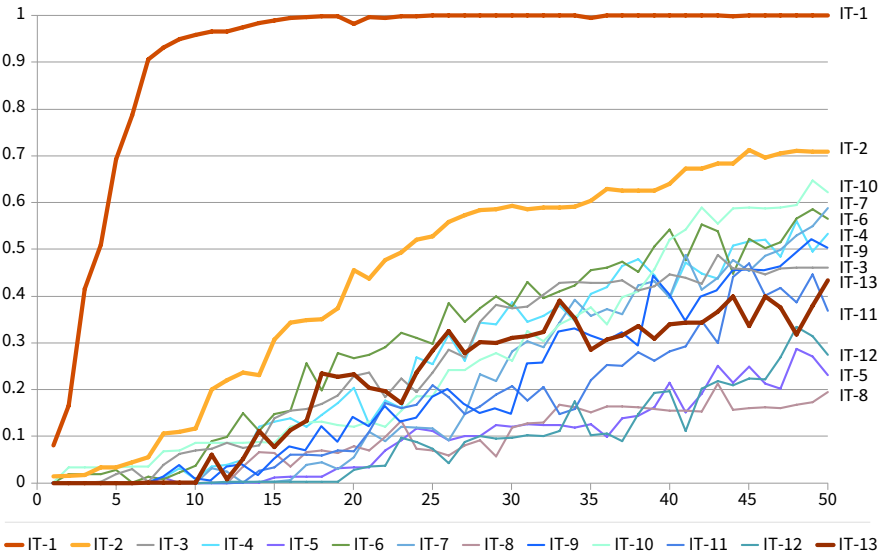


Fig. 16.5 Evolution of the fitness of the best individual of each generation. The results are averages of 30 independent evolutionary runs for each iteration.

unexplainable. In essence, this result means that it is easy for the system to break from his "past" and produce novel imagery.

The initial set of internal images was created by randomly generating genotypes and their corresponding phenotypes. As such, the images of the initial internal dataset did not undergo evolution. By supplying an aesthetic model, and a mechanism that steers evolution towards regions of the search space that were not covered by the initial dataset, however, we are fundamentally changing the nature of the images that the system tends to produce. When confronted by images that are novel, and that probably do not fit into either of the categories (internal or external), the classifier is forced to make a choice, eventually classifying some of these novel images as external. Once such image is found, evolution quickly explores and exploits that type of imagery, leading to the discovery of a high number of images classified as external.

In the second stage, the phenotype diversity mechanisms kicks in, contributing to the discovery of a diversified set of images classified as external. The importance of the phenotype diversity mechanism can be seen from the fact that out of the $38,283$ classified as external, $30,110$ were added to the internal dataset. Thus, only 8173 of the evolved images classified as external, roughly 21%, were considered similar to the ones already in the archive of their corresponding evolutionary runs and were therefore discarded. This result shows that the phenotype diversity mechanism is able to prevent stagnation of the evolutionary runs and convergence to a fixed type of image.

After the explosion of novelty that occurs in the first iteration, the task of the evolutionary engine and, as will be seen, that of the classifier become increasingly harder and an abrupt decrease of productivity is seen. In the course of the 30 generations of the second iteration, the EC engine found 1426 images that were classified as external. Although this is still an impressive number, it pales in comparison with the numbers observed in the first iteration. This increase in difficulty can also be observed from the increase on the average number of generations necessary to find an external image, 18.22, and from the fact that only 22 out of the 30 evolutionary runs were able to find images classified as external. The chart presented in Figure 16.5, concerning the evolution of the fitness of the best individual of each generation across iterations, further highlights the differences in the difficulty of the task of the EC engine in the first and second iterations.

Out of the 1426 external images found in the course of the second iteration, 250 were added to the internal dataset, since the remaining 1176 were considered sufficiently similar to these 250 by our archiving algorithm. This illustrates a well-known fact concerning novelty search algorithms (as defined by Lehman and Stanley (2008)): as optimising fitness becomes harder, it becomes significantly more difficult to find solutions that are both novel and fit. In other words, although the phenotype diversity mechanisms are activated and contribute to the diversity of the population, finding images that are simultaneously novel, in relation to the ones evolved in the course of the evolutionary run, and adequate, i.e. classified as external, becomes increasingly difficult.

As the number of iterations increases, and as the internal dataset becomes larger, one would expect an increasing difficulty in finding images classified as external (and also an increasing difficulty in learning to differentiate between the two sets). Although this tends to be true, it is not always the case. As Figure 16.5 illustrates, although there is a clear differentiation between the lines representing the evolution of fitness for the first two iterations and for the remaining iterations, and although these differences are statistically significant, the same does not happen among the remaining iterations. The explanation for this fact is twofold: (i) the number of images added in each iteration is not sufficient to make the task visibly harder; and (ii) the training of the classifier includes a stochastic component, and so, even if trained with the same datasets, different classifiers may induce different fitness landscapes with different difficulties.

16.5.2 Analysis of the Visual Results

Next we present an analysis of the visual results, i.e. the images, produced in the course of the 13 iterations. The complexity of the setup, and the vast number of images classified as external that were evolved make this analysis particularly hard. Furthermore, and although we have tried to be as objective as possible, the analysis entails a degree of subjectivity that cannot, and perhaps should not, be avoided. We divide this analysis into three subsections, focusing, respectively, on the analysis

Fig. 16.6 Fittest individual from each population of a typical evolutionary run of the first iteration. The image in the upper left corner corresponds to population 0; remaining images in standard reading order.

of the visual results of the first iteration, intermediate iterations, and of the 13th iteration.

16.5.2.1 First Iteration

We begin by trying to convey what happens within each of the 30 evolutionary runs of the first iteration. For this purpose, Figure 16.6 depicts the fittest individual from each of the 50 generations of a typical evolutionary run of the first iteration. As can be observed, the fittest images of the first two generations are quite amorphous. By the third generation, the EC engine finds the first image classified as external. From this point onwards, the phenotype diversity mechanism kicks in, promoting the discovery of images that are, simultaneously adequate, i.e. classified as external, and different from the ones previously evolved in the course of that specific run. This mechanism does not produce immediate effects in terms of the fittest image of the generations, but it prevents the algorithm from converging, and creates the conditions for the discovery, within the evolutionary run, of different images that are also classified as external. Hence, the apparently abrupt changes that can be observed in Figure 16.6 result mainly from a progressive evolutionary process that promotes diversity of the population.

In Figure 16.7, we present a sample of the images classified as external that were evolved in the course of the same run as the one depicted in Figure 16.6. Since we were unable to find a reasonable algorithm for automatically sampling the set of evolved images in a convincing manner, this and other samples presented in this chapter were selected by hand, trying, in all cases, to maximise the diversity of the sample and to make it as representative as possible. As can be observed, the diversity of the populations and of the images classified as external is larger than what Figure 16.6 suggests, showing the adequacy of the phenotype diversity mechanisms.

Figure 16.8 depicts the fittest individual of each of the 30 evolutionary runs of the first iteration. All of these images were classified as external. There are at least three predominant traits: most of the images tend to be dark and with low contrast; several exhibit a star-like shape; and many of them include some sort of noise. In some cases, the contrast is so low that the images appear to be of uniform colour for the human eye; however, a colour adjustment and equalisation operation reveal the hidden structure. Regarding this point, it is relevant to point out that several of the features that serve as input to the ANN are invariant regarding contrast among colours, so these results also highlight the differences in the perception of images between humans and ANNs. It also appears to be safe to state that several of the runs converged to the same type of imagery, which is an expected result. The runs are performed in parallel and the classifier, which ultimately defines the fitness landscape is common to all. Therefore, the fitness landscape has the same local and global optimum, an optima with a larger basin of attraction are bound to be explored more often. Additionally, each evolutionary run has its own archive and no access to the archives of others, and therefore the phenotype diversity mechanisms cannot avoid

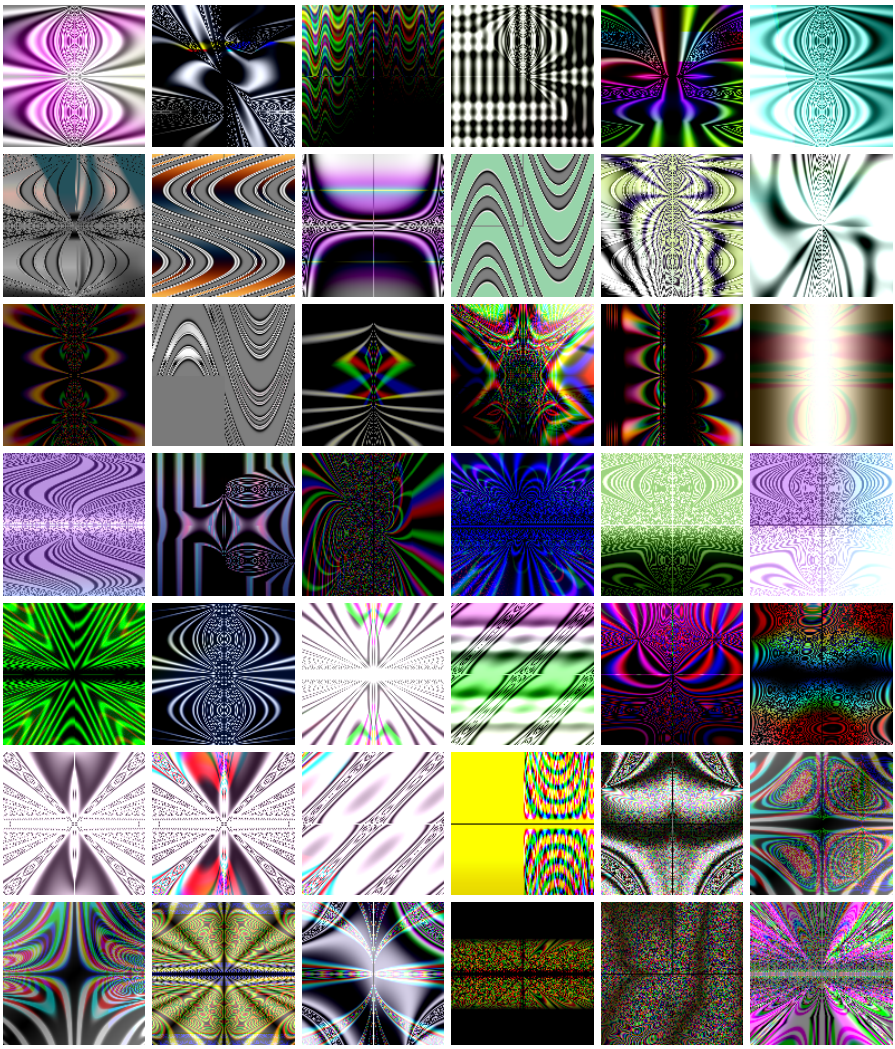


Fig. 16.7 Samples of the images classified as external, generated throughout the course of a single typical evolutionary run of the first iteration.

imagery being explored in other evolutionary runs – they only operate within the production of a specific run.

Figure 16.9 presents a sample of the 38,283 images evolved in the course of the first iteration and classified as external. Obviously, the visual inspection of 38,283 images and the selection of a representative set sufficiently small to present in this chapter would be close to impossible. Nevertheless, we believe that the selected

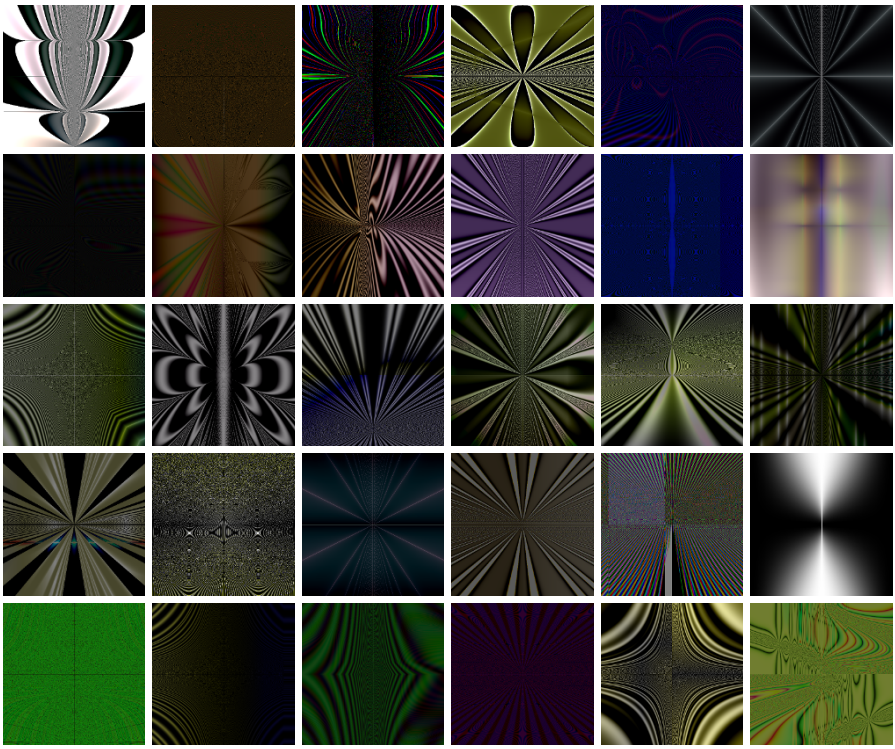


Fig. 16.8 Fittest individuals of the last generation for each of the 30 seeds of the first iteration.

samples illustrate the diversity of images classified as external that were evolved during the course of this iteration.

Based on the results presented, we believe it is safe to claim that the images classified as external are substantially different from the ones belonging to the initial dataset. On the other hand, it is also safe to state that they are substantially different from the external dataset composed of human-made artworks. In a nutshell, the EC engine is producing images that are distinct from both of the initial datasets, and that the classifier, which is forced to classify them into one of these two sets, identified as external. We also believe that it is safe to claim that these images are novel in relation to the ones previously produced by the EC system (i.e. the initial set of internal images), not only from a computational perspective, but also to the human eye.

In our subjective opinion, several of these images are aesthetically interesting and appealing. Considering our background and experience using user-guided evolutionary art systems, which spans more than a decade, it is relevant to make the following observation: these images are, in many ways, similar to the ones we evolved through user-guided evolution in the course of those years. Anecdotal evidence of this fact is that, when confronted with Figure 16.9, one of the authors asked “Why are we

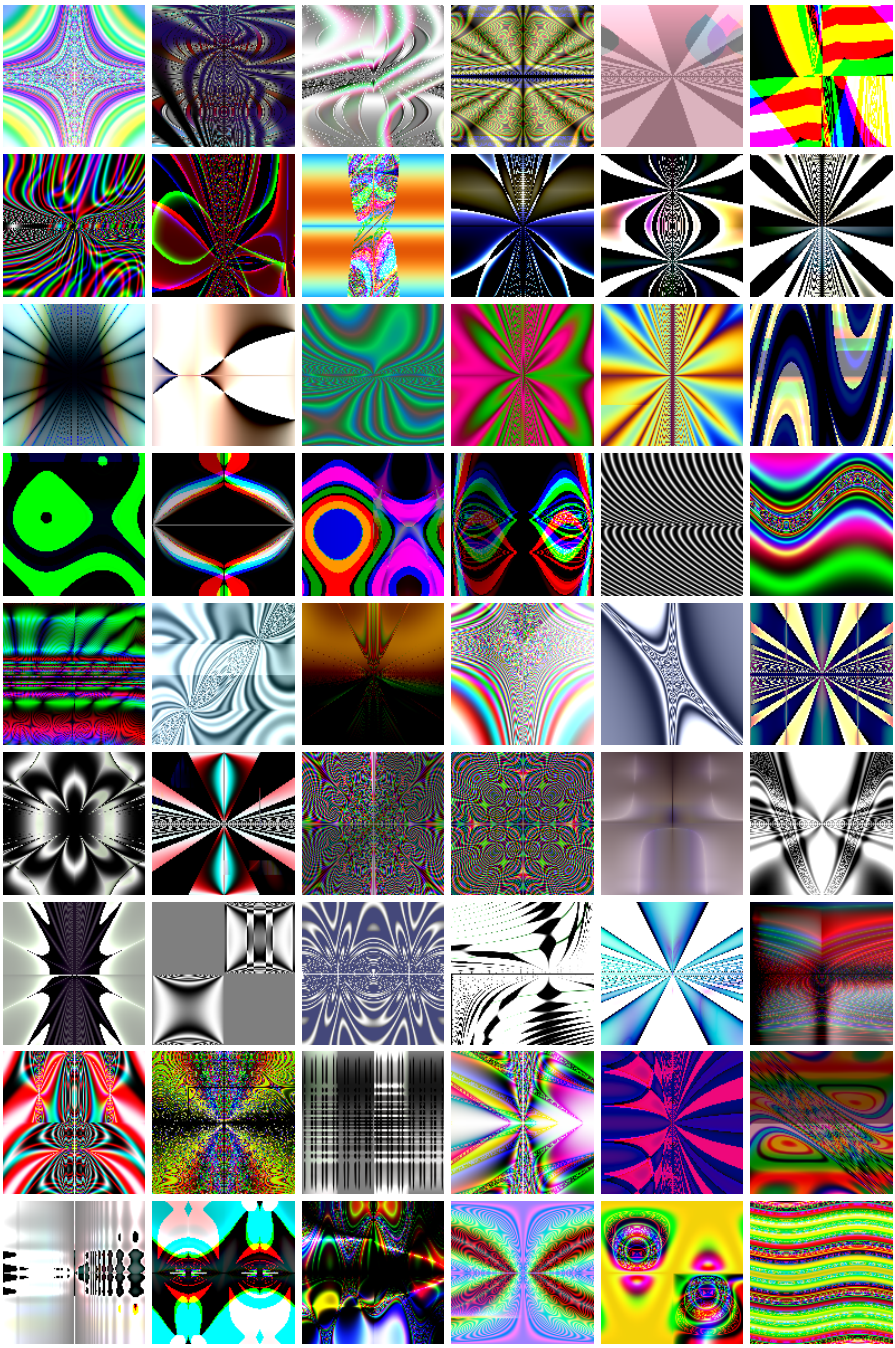


Fig. 16.9 Samples of the images classified as external, generated in the course of the 30 evolutionary runs of the 1st iteration.

including user-guided images?". Proving that this resemblance is real is beyond the scope of the present chapter, nevertheless, even without strong evidence to make this claim, we consider this one of the most unexpected, and possibly relevant, results in this chapter.

16.5.2.2 Intermediate Iterations

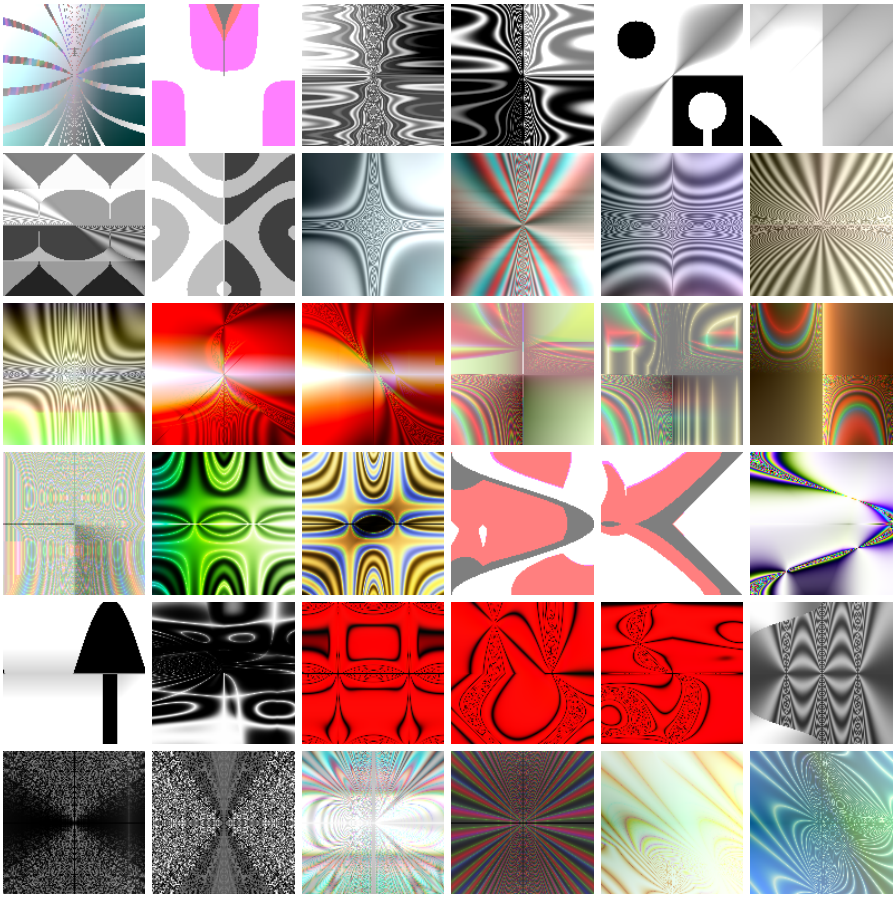


Fig. 16.10 Samples of the images classified as external, generated in the course of the 30 evolutionary runs of the second iteration.

In this subsection we give an overview of the visual results obtained in the second to the 12th iterations. These results are illustrated by the samples of the images classified as external presented in Figures 16.10-16.20. It is important to remember

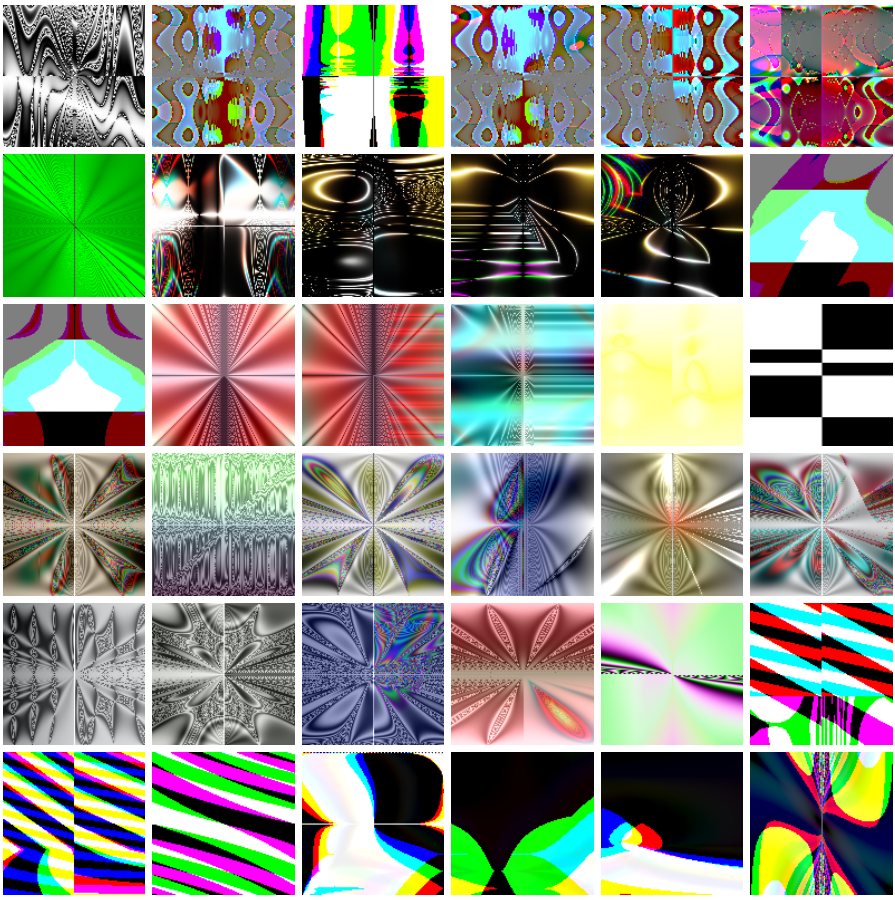


Fig. 16.11 Samples of the images classified as external, generated in the course of the 30 evolutionary runs of the third iteration.

that, in most cases, for each of the images presented in the figures a significant number of images of similar nature were evolved in the corresponding run.

Rather than performing a detailed analysis, we will focus on highlighting some of the most striking results obtained in each iteration, identifying, whenever possible, trends that emerge in several runs and that, as such, represent optima with a large basin of attraction for the classifier being used in that particular interaction.

In the course of the second iteration the EC engine evolved 1436 images classified as external. These images resulted from 22 of the 30 runs. The images presented in the figures are ordered by evolutionary run. Two similar images presented side by side typically indicate that they were evolved in the same run, and similar images that are not adjacent to each other typically indicate the rediscovery of the same type of imagery in two different runs.

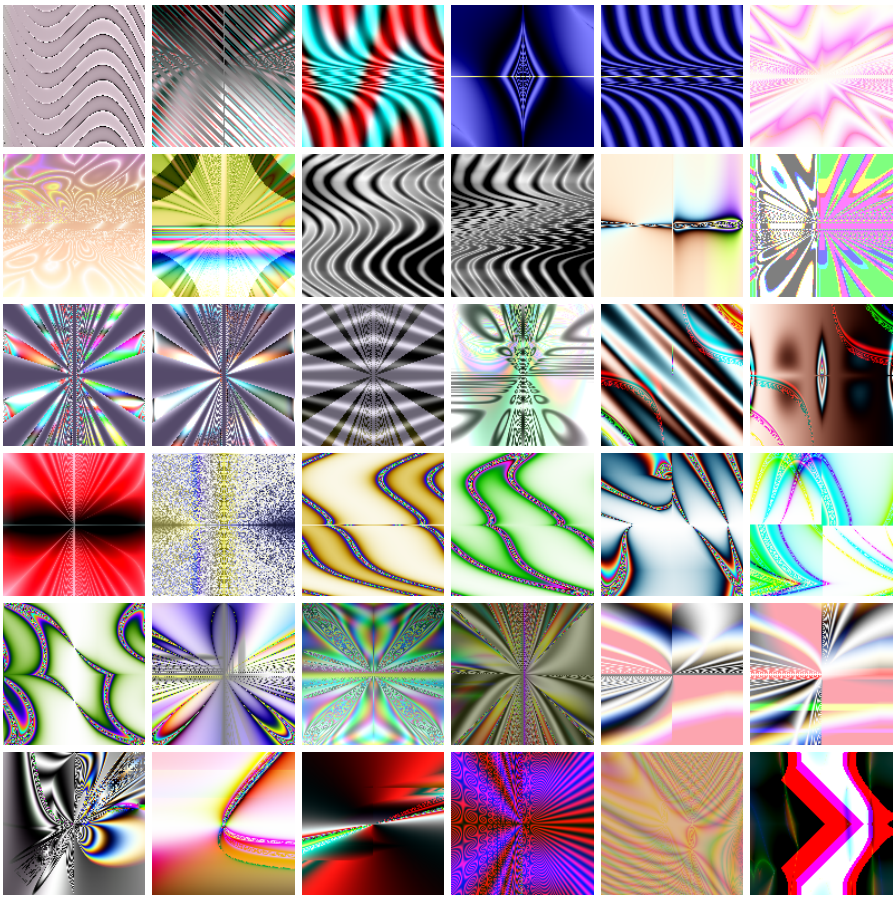


Fig. 16.12 Samples of the images classified as external, generated in the course of the 30 evolutionary runs of the fourth iteration.

A brief scrutiny of the images presented in Figure 16.10 reveals that most of the evolutionary runs converged to different imagery, but also shows the recurrence of some themes. Among these, we highlight the striped star-like shapes, which also emerged in the first iteration, and which continue to be present, although rendered in a different style. One of the interesting results concerns the evolution of several minimalistic images (e.g. the two rightmost images in the first row and the leftmost images in the fifth row), which occurs in several runs. Although they appear minimalistic, this type of image is particularly hard to evolve, and their simplistic nature contrasts with the size of their genotype. In fact, an inspection of the learning process after the second iteration appears to indicate that the emergence of these images is deeply related to the presence in the initial dataset of external imagery that are also minimalistic and monochromatic (see Sect. 16.5.3). In fact the use of a reduced

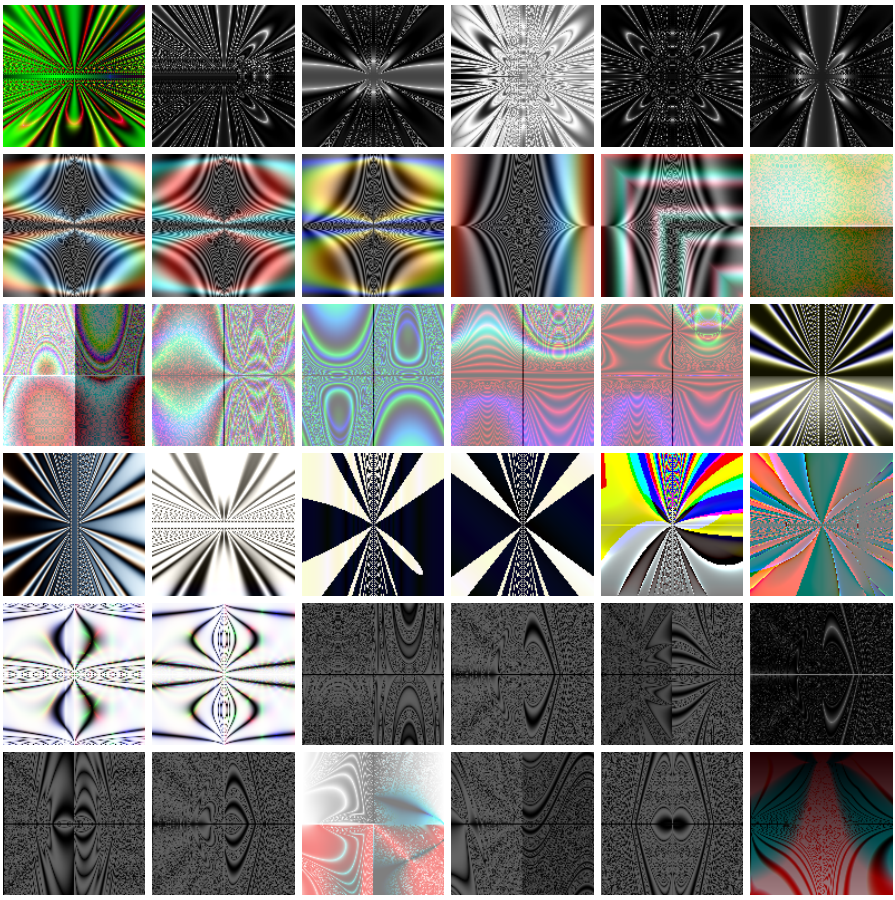


Fig. 16.13 Samples of the images classified as external, generated in the course of the 30 evolutionary runs of the fifth iteration.

colour palette occurs in several of the evolutionary runs. This is consistent with the colour schemes used in many of the images belonging to the external dataset and contrasts with with the typical imagery produced by the EC engine. More importantly, considering the nature of this chapter, the appearance of the evolved images classified as external appears, in most cases, to be different from that of the initial dataset of external images and from that of the images evolved in the course of the first iteration.

Analysing the images produced in the course of the third iteration, of which a sample is presented in Figure 16.11, one can observe the same overall patterns: most runs tend to converge to different types of images; most evolved novel imagery in relation to the previous production of the system; and there are some recurring themes, namely the star-like images, which are "rendered" in different styles. The

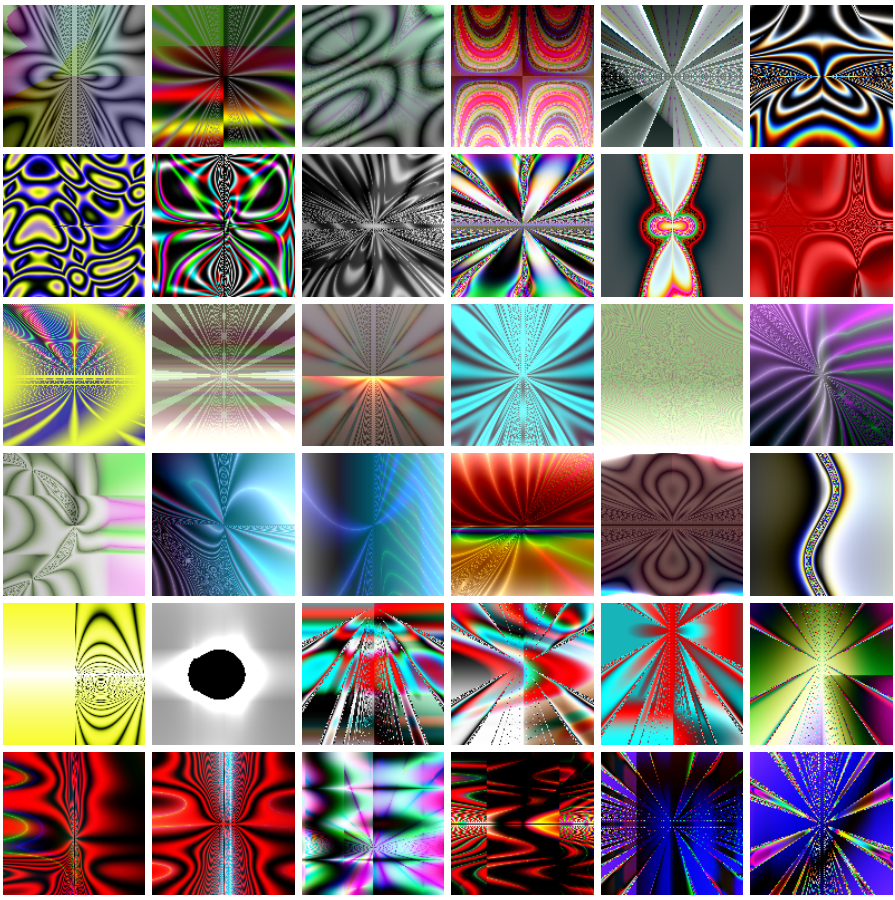


Fig. 16.14 Samples of the images classified as external, generated in the course of the 30 evolutionary runs of the sixth iteration.

emergence of images with strong and contrasting colours (magenta, green, yellow, white, black) occurs in several evolutionary runs. This type of imagery is highly atypical of the EC engine and matches the chromatic characteristics of several of the artworks in the external set.

As we will see when analysing the results of other iterations, the emergence of graphic elements such as lines, points and planes, also characterises some of the evolved images. Although these are usually considered as graphic primitives for humans, the EC engine has no explicit way of creating such elements. Hence, their emergence is deeply linked to the fitness landscape induced by the classifiers.

Much of what was stated regarding the images evolved in the third iteration also applies to the ones evolved in the fourth (see Figure 16.12). Many of the images are characterised by the emergence of organic lines and planes. Others appear to be

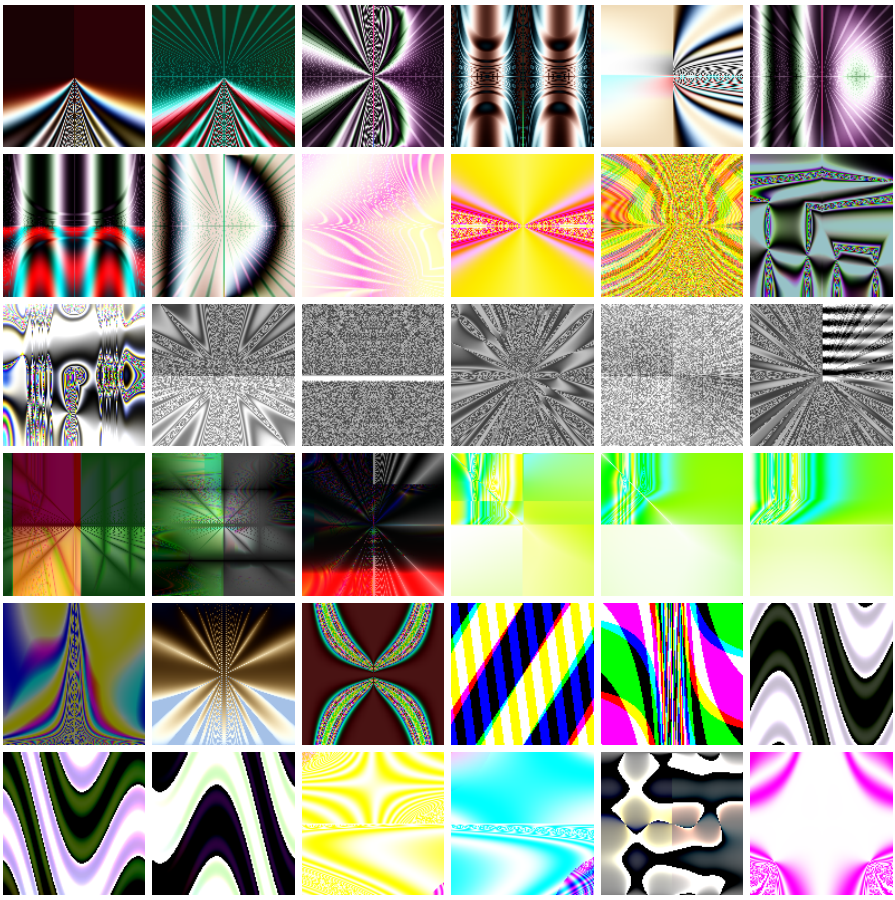


Fig. 16.15 Samples of the images classified as external, generated in the course of the 30 evolutionary runs of the seventh iteration.

composed of multiple layers with transparencies (e.g. the leftmost image in the third row).

In the fifth iteration, the EC engine experienced difficulties in finding images classified as external. Only 10 of the 30 evolutionary runs found such images and, on average, these took 29.3 generations to evolve. In total, 366 images classified as external were evolved, a number that was reduced to 57 by our archiving algorithm. For these reasons, the diversity of the images presented in Figure 16.13 is not as large as for previous iterations. The feature common to all of these images is the presence of noise patterns. It is also interesting to notice that a vast percentage of the images are monochromatic and have intricate detail. In several of these cases (e.g. the black and white images in the first and last row) the lines are discontinuous, in

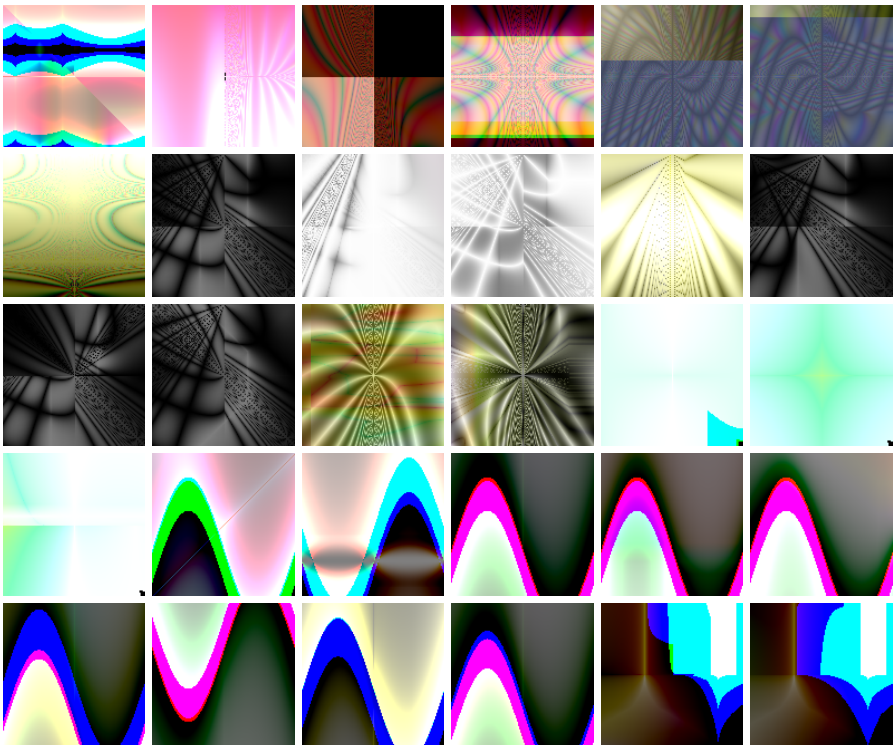


Fig. 16.16 Samples of the images classified as external, generated in the course of the 30 evolutionary runs of the eighth iteration.

the sense that they emerge from the arrangement of several white or grey dots that are not actually connected.

Although the productivity of the AA during the fifth iteration was not high, the addition of these images to the internal dataset, coupled with the removal of some of the external images (see Section 16.5.3), appears to cause profound changes in the classifier. There is a burst of productivity in the course of the sixth iteration, 1105 images, of which 433 are added to the archive, a number that is surpassed only by the first iteration. As Figure 16.14 illustrates, this burst of productivity coincides with a change in style in comparison with the previous iterations. This sudden increase in productivity can be explained by the performance of the classifier, and will be discussed in Sect. 16.5.3.

Productivity decreases during the seventh iteration (Figure 16.15) and reaches an all time low in the eighth iteration (Figure 16.16). Generally speaking, one can state that the images classified as external that are evolved in the course of the seventh iteration correspond to variations in the style of themes already explored in previous iterations, almost as if the AA has further refined and included additional detail in

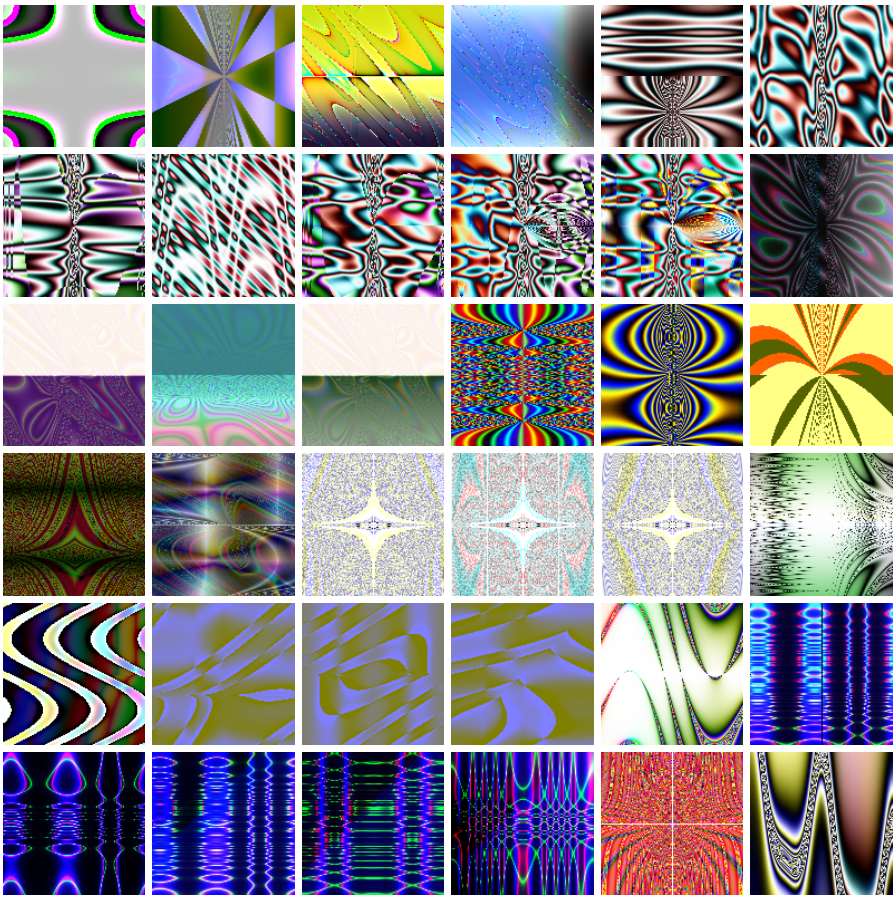


Fig. 16.17 Samples of the images classified as external, generated in the course of the 30 evolutionary runs of the ninth iteration.

previously explored images. The ones evolved in the eighth iteration appear, to our eyes, to be of the same style as images evolved in some of the previous iterations. The classifier is not able, even during training, to fully discriminate between the internal and external datasets. As previously explained, this opens the door for the repetition of styles and imagery that was not sufficiently explored in previous iterations. Our analysis indicates that this is what happened in the course of the eighth iteration, the AA explored styles that, although already present, had not been sufficiently explored. As the cardinality of such images increases the classifier system is forced to recognise such images as internal and, therefore, the EC engine will no longer be able to explore them in future iterations.

As previously, the changes in the datasets gave rise to a classifier that based its assessments on different premises, inducing a different fitness landscape, which

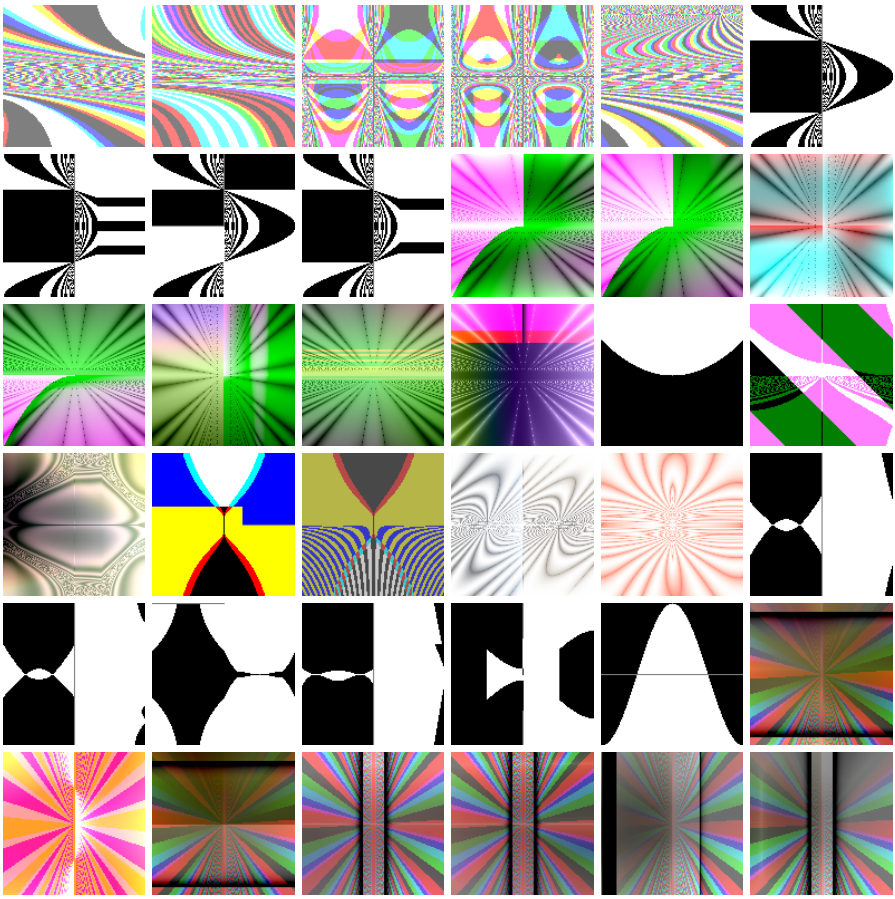


Fig. 16.18 Samples of the images classified as external, generated in the course of the 30 evolutionary runs of the 10th iteration.

happens to be more prone to evolution. In the ninth iteration, the EC engine evolved a total of 422 images, of which 115 were archived, finding images classified as external in 21 of the 30 runs. As can be observed by inspecting Figure 16.17, there is a mixture of new and old themes and styles. Interestingly, several images that are evocative of landscapes (the three leftmost images in the third row) were evolved.

The 10th iteration was one of the most productive ones in terms of the total number of images classified as external, 692, but of these only 62, less than 10%, made it to the archive. As it can be observed in Figure 16.18, several runs converged to the same type of imagery, reducing the overall diversity and productivity of the set. Visually, we can identify three main styles which emerge in several runs: the black and white minimalistic images, images that appear to have a white transparent layer (e.g. the five leftmost images in the first row, but also the two rightmost in the last

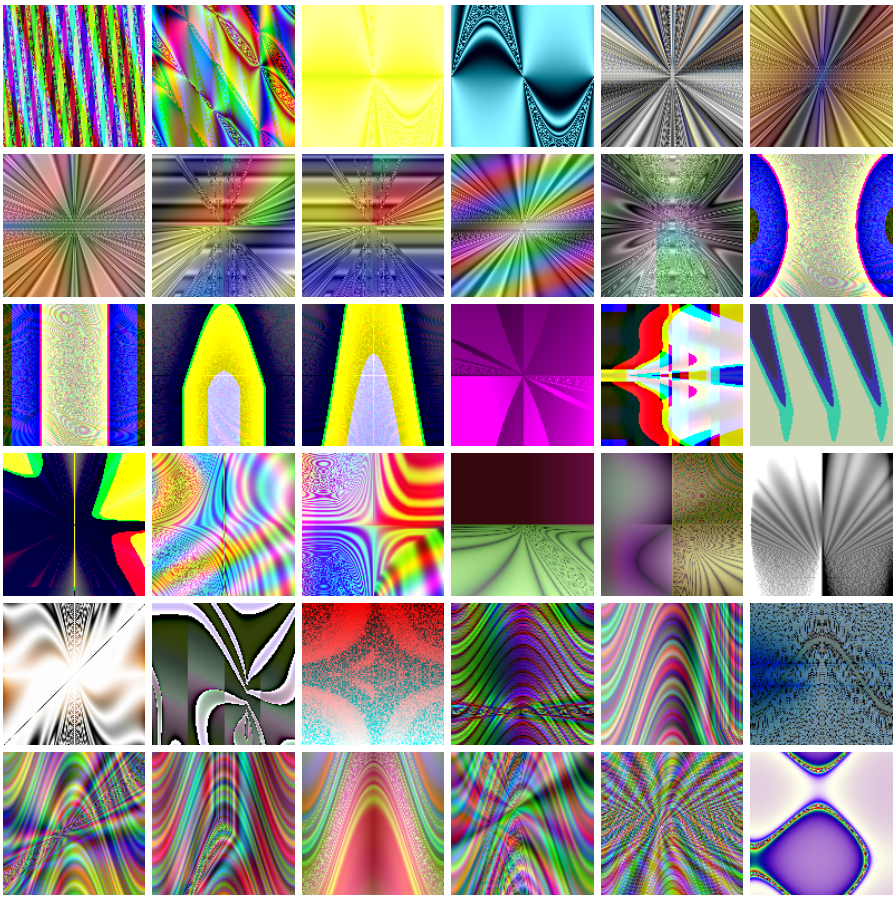


Fig. 16.19 Samples of the images classified as external, generated in the course of the 30 evolutionary runs of the 11th iteration.

row), and images exploring a combination of magenta and green. As in several of the previous iterations, the star-like shape continues to be one of the favorite "themes" of the AA.

The lack of diversity in the 10th iteration contrasts with the visual diversity in the 11th. Although only 374 images classified as externals were found, 126 of these images were archived. On average it took 32.27 generations to find the first image classified as external. Although there is some stylistic agreement among several evolutionary runs (see Figure 16.19), the overall diversity is significantly higher than in the previous iteration. The purely black and white images disappear from this iteration onwards, likely owing to the combination of two factors: the inclusion of several of these images in the internal dataset and, most importantly, the removal of a large number of strictly black and white images from the external dataset.

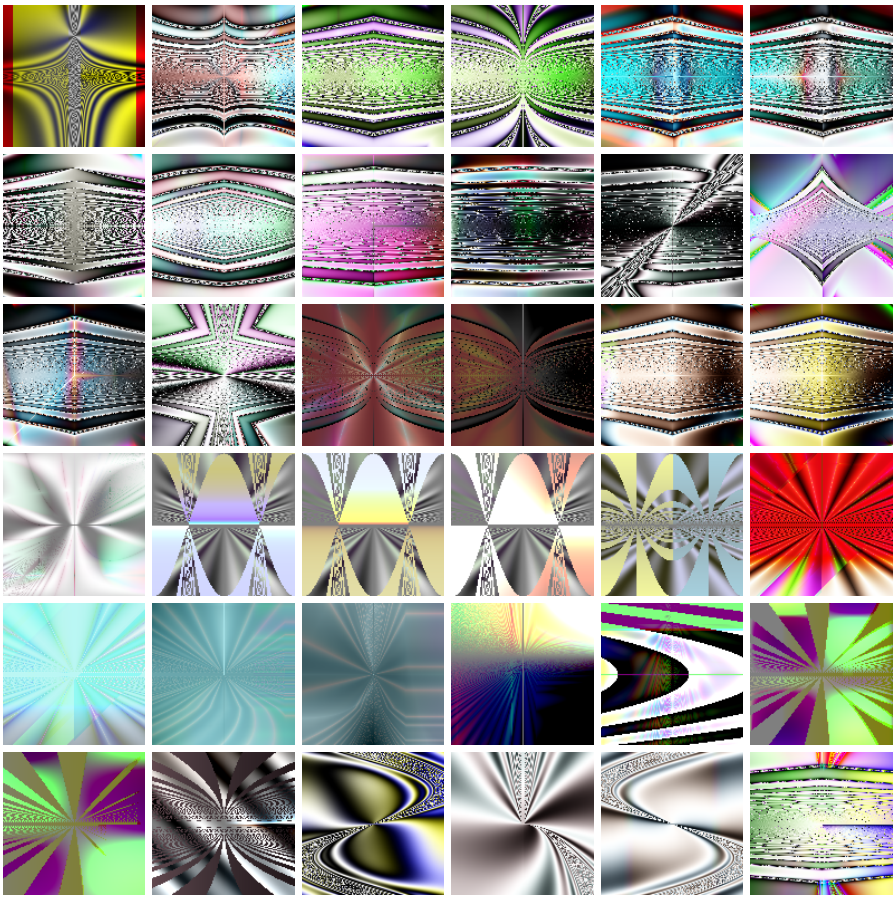


Fig. 16.20 Samples of the images classified as external, generated in the course of the 30 evolutionary runs of the 12th iteration.

The 12th iteration was among the least productive ones; only 267 images classified as external were found, and only 11 of the 30 runs found such images. In spite of this lack of productivity, visible in Figure 16.20, some of the evolutionary runs were able to find novel imagery that contrasts in terms of both style and theme from the previous artistic production of the system.

16.5.2.3 Thirteenth Iteration

The 13th, and the last iteration presented in this chapter, corresponds to a burst of novelty and productivity in the system. Although a similar burst occurred in the sixth iteration, the nature of this burst appears significantly different. In this case,

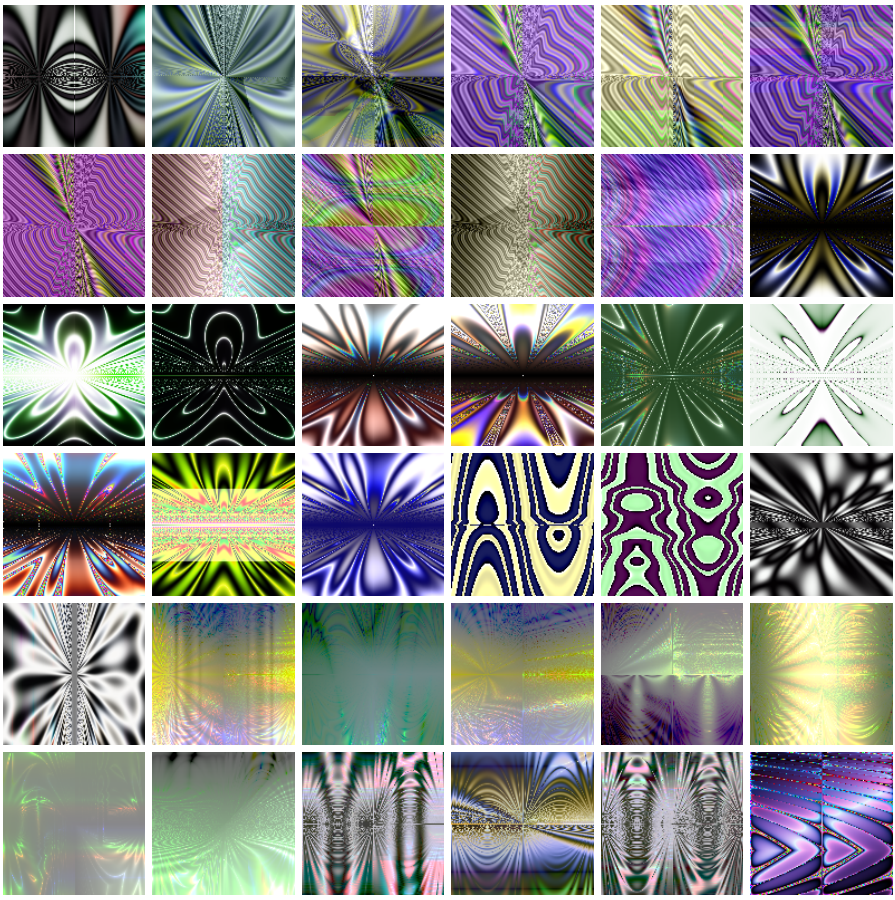


Fig. 16.21 Samples of the images classified as external, generated in the course of the 30 evolutionary runs of the 13th iteration.

the increased productivity is coupled with significant stylistic variations and may be seen as a moment where the AA actually broke the mould.

As we will see in the next section, while the increase of productivity in the sixth iteration seems to be linked to shortcomings of the classifier, here it appears to be linked to significant changes in the way the classifier system differentiates between the classes of internal and external imagery. Thus, while in the other intermediate iterations the AA seems to be making minor stylistic variations on images that it has already produced, and opportunistic exploitations of shortcomings of the classifier, what happens in the 13th iteration seems to be rather different, resulting from profound changes in the aesthetic model, caused by the cumulative revision of the internal and external set. Making an analogy, this can be seen as an "Eureka"

moment, where the system discovers substantially different styles, expanding and enriching the range of its artistic production.

As it can be observed from the sample of images presented in Figure 16.21, although there are some recurring themes, the detail of execution of the images of the 13th generation classified as external is a lot higher than in previous iterations. The images seem to be more elaborate, detailed and refined when compared with previous iterations. At the same time, some novel ornamentation techniques, such as the one depicted in the three rightmost images in the first column, have been discovered, and some novel themes seem to emerge. The exploration of light (see, e.g. the leftmost image in the third row and the rightmost image in the fifth row) also emerges as a visible and distinctive trait.

We considered the images resulting from the first iteration to be comparable to the evolved through user-guided evolution. Although, in fairness, the same could be said for a significant portion of the images evolved in the course of the 13th iteration, it is equally fair to state that some of the runs created imagery that was stylistic dissimilar to what we have evolved either through user-guided evolution, or other means. Thus, many of these images strike us not only as novel in relation to the previous artistic production of the AA, but also as novel and surprising in relation to our own experience and production.

16.5.3 Training of the Classifiers

In this subsection we give an overview of the results pertaining the training of the classifier. As previously mentioned (see Section 16.4.1), when an iteration is concluded the images classified as external that are evolved in the course of the iteration, are gathered together by the supervisor and added to the internal dataset. This is followed by several training attempts, which may imply removing images from the external dataset. Training is concluded when no external images are classified as being internal. The existence of internal images classified as external implies that the EC engine may revisit previous styles, but, when this occurs, the consequent increase in the number of images in those styles present in the internal dataset will eventually force the classifier to recognise such images as internal.

Table 16.6 presents a summary of some pertinent statistics regarding the training phase. It details, per iteration: the number of training attempts necessary to reach a classifier without false internals (attempts); the total number of false externals identified in the course of these attempts (False Externals); the number of false externals and internals after the training attempts are concluded (CS False External and CS False Internal, respectively), which reflects the ability of the classifier that is going to be used to guide the following iteration to discriminate between the sets; and the sizes of the external and internal datasets after training is concluded (Total External and Total Internal).

As can be observed, the training of the classifier in the first iteration required two attempts. One external image, a black and white photograph of a detail of a painting,

Table 16.6 Statistics regarding the training of the classifiers in each iteration in terms of: number of attempts, false internals and externals during training, false externals after training, and size of the internal and external and internal dataset after training.

		During Training Cycles		After Training Attempts			
Iteration	Attempts	False External	False Internal	CS False Externals	CS False Internals	Total Externals	Total Internal
initial	2	1	1	1	0	26238	26239
1	7	23	68	0	0	26170	56349
2	4	52	20	3	0	26150	56599
3	7	41	67	3	0	26083	56794
4	6	7	11	3	0	26072	56972
5	2	40	18	28	0	26054	57029
6	2	7	18	3	0	26036	57462
7	3	11	248	3	0	25788	57593
8	3	23	31	0	0	25757	57624
9	3	11	8	5	0	25749	57739
10	3	126	6	5	0	25743	57801
11	8	145	47	4	0	25696	57927
12	6	99	63	7	0	25633	58028
13	4	88	64	3	0	25589	58380

was removed from the external dataset. Unfortunately, owing to copyright issues this and other external images cannot be reproduced in the chapter. After the second attempt, no external images were classified as internal, but one internal image, a black and white star-like shape, was deemed as external. The existence of this image, and the difficulty in classifying it, may at least partially explain the recurrence of such a theme in several iterations.

As previously mentioned, the first iteration generated a large and varied set of images. As a consequence, 30,110 images were added to the internal dataset and the training of the classifier that guided the second iteration took a significantly larger number of attempts; in total 68 external images were removed. These were, mostly, black and white engravings, and quite interestingly, some images of mathematical objects, an artwork by M. C. Escher, which also had a mathematical appearance, and a cartoon image. Concerning the engravings, we believe that these images were removed for two main reasons: (i) at the resolution that the feature extractor processes these images, they could easily be confused with images produced by the EC engine; (ii) these images tend to be atypical in relation to the other images belonging to the external dataset, which makes them harder to classify. Concerning the images of mathematical objects, these seem to be computer generated and therefore the confusion with the images produced by the AA is natural. After the removal of these images, the classifier was able to fully distinguish between the two sets.

The same overall trend occurs in iterations 2 and 3, although the number of attempts varies (4 and 7, respectively) the types of external images excluded were the same, including engravings, Escher artworks, black and white drawings, photographs of sculptures, and minimalistic paintings (some by Kazimir Malevich). The reasons for their misclassification are the same: they are either atypical in relation with to

rest of the external set or, confusable with computer generated imagery, i.e., similar in style to the images the EC engine is prone to create.

The fourth generation provoked few changes in the external dataset, removing only 11 images. While 10 of these were black and white drawings, the remaining one is notable since it was the first Mondrian removed from the external set. The images produced in the fifth generation, provoked the exclusion of 18 images from the external dataset, among which were two by Matisse and two by Escher. The most relevant issue concerning the training subsequent to the fifth iteration is that the resulting classifier, which guide evolution in the sixth iteration, misclassifies 28 internal images. This gives the EC engine a large degree of freedom to explore previously visited imagery, which explains the burst of productivity observed in the sixth iteration. The exploitation of these shortcomings leads to the generation of images that, once added to the internal dataset prevent their future exploitation.

In the seventh iteration, a total of 248 external images were removed from the external dataset, these included black and white engravings, several Escher artworks (14 to be precise), several Mondrian paintings, and numerous line drawings. The eighth iteration, the least productive of all, provoked the removal of 31 external images, which tended to be of the same type as the ones previously identified. After these removals, the classifier was again able to fully distinguish between the two sets.

The ninth and 10th iterations caused the removal of few external images, 8 and 6, respectively. This confirms our previous observation that, although these iterations were productive, they were not particularly fruitful in terms of the novelty of the evolved imagery. As mentioned previously, although 692 images classified as external were evolved in the 10th iteration, only 62 of these made it to the archive.

These numbers contrast with the ones for the 11th and 12th iterations, where 47 and 63 respectively, were deleted. These included several Picasso, Dalí, Paul Klee, Mondrian, and Mark Rothko paintings, as well as several line drawings. The lack of texture, at low resolution, appears to be the trait that links these paintings.

As mentioned previously, in our opinion, the thirteenth iteration is different from the others in the sense that it corresponds to a pronounced shift in style. Hence, it is particularly interesting to inspect what kind of changes to the external dataset the evolved imagery induces. In total 64 external images were removed. In previous iterations most of the removed images were black and white drawings or engravings but not the case in the 13th generation; only ten of the deleted images were black and white. The remaining images are Renascence style paintings (unfortunately, they do not include the Mona Lisa), two Van Gogh paintings, three by Kandinsky and one by Miró. Quite interestingly, four of Monet's paintings of the Waterloo Bridge, a theme that is present in several of his artworks, were also removed. In this case it was possible, and quite easy, we might add, to identify the evolved images that promoted the confusion between internal images and these artworks. Some of them are depicted in the bottom two rows of Figure 16.21, and we believe that the reader will also understand why, in the eyes of the classifier, they could easily be confused.

16.6 Conclusions and Future Work

In this chapter we have presented an artificial artist that is characterised by its permanent quest for novelty. The system is composed of two main modules: a generator and an evaluator. The role of the generator is played by an expression based evolutionary engine and that of the evaluator by an artificial neural network. The network is trained to discriminate between the images produced by the evolutionary engine and a set of famous artworks. The fitness of the images that are evolved depends on the output of the evaluator, promoting the discovery of images that the network classifies as external. In each iteration of the framework we perform 30 evolutionary runs. When these are concluded, the relevant misclassified images are added to the set representing the production of the AA and the neural network is retrained.

For the above reasons, the approach promotes and explores competition between the generator and evaluator. From a theoretical standpoint – assuming that the evolutionary engine and the artificial neural network are adequate and always able to cope – the iterative expansion of the internal set leads, necessarily, to change since the evolutionary algorithm is forced to explore new paths. Moreover, assuming that a sufficiently large number of iterations performed and that both systems cope, a convergence to the aesthetic model (or models) implicitly defined by the set of external images, which provides an aesthetic reference for the artistic production of the AA, is bound to eventually occur.

To increase the diversity within evolutionary runs, and prevent their early convergence and stagnation, we include mechanisms to promote the phenotype diversity of the populations. This implies taking two criteria into account when performing tournament selection: the adequacy of the image (which results from the output of the neural network) and its diversity in relation to the images produced in the course of the same run.

The analysis of the experimental results confirms the adequacy and potential of the approach, revealing that the system is able to consistently produce novel imagery that arguably has aesthetic merit. Hence, we consider that we successfully developed a creative system that is able to learn, create and innovate in an entirely autonomous way.

The experimental results indicate that the images produced in the course of the first iteration of the framework are similar to those produced by expression-based interactive evolutionary art systems, where the role of the evaluator is played by a human. Analysing the results, we consider that the behaviour and production of the system during the first 12 iterations can be considered as e-creative. We consider that what happens during the 13th generation is significantly different and goes beyond e-creativity. In this case, the system made a qualitative and substantial change in terms of both production and aesthetic model, thus breaking the mould. We have put forward the hypothesis that this behaviour can be seen as a limited case of h-creativity in the sense that the system produced images that appear to be different from those previously attained by evolutionary means and of t-creativity, in the sense that the

changes appear to be related to deep changes in the aesthetic model and, therefore, to a profound transformation of the search space.

Future work will focus on two main aspects: further improvement of the framework, and additional testing, namely testing some of the hypothesis that we have made. Concerning the refinement of the framework, we consider that the framework can only be fully assessed once it has been pushed to its limits. We have presented the results of thirteen iterations, but further ones are still being performed. Although it is impossible to predict, the number of iterations necessary to provoke the collapse of one of the modules, that number is likely to be large. By taking the framework to the point of collapse, we hope to gain additional insights regarding the limitations of the evolutionary engine and, specially, of the classifier. Such insights will guide future developments. The inclusion of a curator module, which would select a small set of artworks from the production of the AA, would also be a valuable addition to the framework. From a conceptual standpoint, this confers to the system a degree of introspection and self-analysis that it currently lacks. From a practical perspective, it would avoid the need of hand-picking representative examples, which may induce a bias in the presentation of the experimental results. Concerning testing, we are particularly interested in finding out wether the images that were evolved in the course of the first interaction are indeed similar, to humans and computers, to those resulting from user-guided evolution. For that purpose, we are gathering a set of user-evolved images and will test wether if humans and computers are able to discriminate between them.

Acknowledgements The ConCreTe project acknowledges financial support from the Future and Emerging Technologies (FET) programme within the Seventh Framework Programme for Research of the European Commission, under FET grant number 611733. This research was also partially funded by the Fundação para a Ciência e Tecnologia (FCT), Portugal, under grant SFRH/BD/90968/2012.

References

Atkins, D. L., Klapaukh, R., Browne, W. N., & Zhang, M. (2010). Evolution of aesthetically pleasing images without human-in-the-loop. In *IEEE Congress on Evolutionary Computation* (pp. 1–8). IEEE.

Baluja, S., Pomerlau, D., & Todd, J. (1994). Towards automated artificial evolution for computer-generated images. *Connection Science*, *6*(2), 325–354.

Barnsley, M. F. (1993). *Fractals Everywhere* (2nd). Cambridge, MA: Academic Press.

Boden, M. (2004). *The Creative Mind: Myths and Mechanisms*. Routledge.

Correia, J. (2009). *Evolutionary Computation for Assessing and Improving Classifier Performance* (MSc Dissertation, Department of Informatics Engineering, University of Coimbra).

Correia, J., Machado, P., Romero, J., & Carballal, A. (2013a). Evolving figurative images using expression-based evolutionary art. In *Proceedings of the 4th International Conference on Computational Creativity, ICCC 2013* (pp. 24–31).

Correia, J., Machado, P., Romero, J., & Carballal, A. (2013b). Feature selection and novelty in computational aesthetics. In P. Machado, J. McDermott, & A. Carballal (Eds.), *Evolutionary and Biologically Inspired Music, Sound, Art and Design, 2nd International Conference, EvoMUSART 2013, Vienna, Austria, April 3-5, 2013. Proceedings, Theoretical Computer Science and General Issues* (Vol. 7834, pp. 133–144). Lecture Notes in Computer Science. Springer.

Datta, R., Joshi, D., Li, J., & Wang, J. Z. (2008). Image retrieval: Ideas, influences, and trends of the new age. *ACM Computing Surveys*, *40*, 5:1–5:60. doi:http://doi.acm.org/10.1145/1348246.1348248

Datta, R., Joshi, D., Li, J., & Wang, J. Z. (2006). Studying aesthetics in photographic images using a computational approach. In *Computer Vision – ECCV 2006, 9th European Conference on Computer Vision, Part III* (pp. 288–301). LNCS. Graz, Austria: Springer.

den Heijer, E. (2012). Evolving art using measures for symmetry, compositional balance and liveliness. In *Proceedings of the 4th International Joint Conference on Computational Intelligence (IJCCI)* (pp. 52–61).

den Heijer, E., & Eiben, A. E. (2010). Comparing aesthetic measures for evolutionary art. In Cecilia Di Chio et al. (Ed.), *Applications of Evolutionary Computation, EvoApplications 2010: EvoCOMNET, EvoENVIRONMENT, EvoFIN, EvoMUSART, and EvoTRANSLOG, Istanbul, Turkey, April 7-9, 2010, Proceedings, Part II, Theoretical Computer Science and General Issues* (Vol. 6025, pp. 311–320). Lecture Notes in Computer Science. doi:10.1007/978-3-642-12242-2-32

Ekárt, A., Joó, A., Sharma, D., & Chalakov, S. (2012). Modelling the underlying principles of human aesthetic preference in evolutionary art. *Journal of Mathematics and the Arts*, *6*(2-3), 107–124. doi:10.1080/17513472.2012.679489

Faria, J., Bagley, S., Ruger, S., & Breckon, T. (2013). Challenges of finding aesthetically pleasing images. In *14th International Workshop on Image Analysis for Multimedia Interactive Services (WIAMIS), 2013* (pp. 1–4). IEEE.

Goshtasby, A. A. (2012). Similarity and dissimilarity measures. In A. A. Goshtasby (Ed.), *Image Registration: Principles, Tools and Methods* (pp. 7–66). doi:10.1007/978-1-4471-2458-0_2

Greenfield, G. (2002a). Color dependent computational aesthetics for evolving expressions. In R. Sarhangi (Ed.), *Bridges: Mathematical Connections in Art, Music, and Science; Conference Proceedings 2002* (pp. 9–16). Winfield, KS: Central Plains Book Manufacturing.

Greenfield, G. (2002b). On the co-evolution of evolving expressions. *International Journal of Computational Intelligence and Applications*, *2*(1), 17–31.

Greenfield, G. (2003). Evolving aesthetic images using multiobjective optimization. In B. McKay et al. (Eds.), *Congress on Evolutionary Computation, CEC 2003* (Vol. 3, pp. 1903–1909). Canberra: IEEE Press.

Greenfield, G. (2005). Evolutionary methods for ant colony paintings. In Rothlauf, Franz et al. (Ed.), *Applications of Evolutionary Computing, EvoWorkshops 2005: EvoBIO, EvoCOMNET, EvoHOT, EvoIASP, EvoMUSART and EvoSTOC, Theoretical Computer Science and General Issues* (Vol. 3449, pp. 478–487). LNCS. Springer.

Hall, M., Frank, E., Holmes, G., Pfahringer, B., Reutemann, P., & Witten, I. H. (2009). The WEKA data mining software: An update. *SIGKDD Explorations Newsletter*, *11*(1), 10–18. doi:10.1145/1656274.1656278

Kowaliw, T., Dorin, A., & McCormack, J. (2009). An empirical exploration of a definition of creative novelty for generative art. In *Proceedings of the 4th Australian Conference on Artificial Life: Borrowing from Biology, ACAL 2009, Lecture Notes in Artificial Intelligence* (Vol. 5865, pp. 1–10). doi:http://dx.doi.org/10.1007/978-3-642-10427-5_1

Koza, J. R. (1992). *Genetic Programming: On the Programming of Computers by Natural Selection*. Cambridge, MA: MIT Press.

Lehman, J., & Stanley, K. O. (2008). Exploiting open-endedness to solve problems through the search for novelty. In *Proceedings of the 11th International Conference on Artificial Life (ALIFE XI)*, Cambridge, MA: MIT Press.

Li, C., & Chen, T. (2009). Aesthetic visual quality assessment of paintings. *IEEE Journal of Selected Topics in Signal Processing*, *3*(2), 236–252. doi:10.1109/JSTSP.2009.2015077

Li, Y., Hu, C., Chen, M., & Hu, J. (2012). Investigating aesthetic features to model human preference in evolutionary art. In P. Machado, J. Romero, & A. Carballal (Eds.), *Evolutionary and Biologically Inspired Music, Sound, Art and Design - 1st International Conference, EvoMUSART 2012, Málaga, Spain, April 11-13, 2012. Proceedings, Theoretical Computer Science and General Issues* (Vol. 7247, pp. 153–164). Lecture Notes in Computer Science. Springer.

Liapis, A., Yannakakis, G. N., & Togelius, J. (2013). Sentient sketchbook: Computer-aided game level authoring. In *Proceedings of the 8th International Conference on the Foundations of Digital Games, FDG 2013, Chania, Crete, Greece, May 14-17* (pp. 213–220).

Machado, P., & Cardoso, A. (1997). Model proposal for a constructed artist. In N. Callaos, C. Khoong, & E. Cohen (Eds.), *First World Multiconference on Systemics, Cybernetics and Informatics, SCI97/ISAS97* (pp. 521–528). Caracas, Venezuela.

Machado, P., & Cardoso, A. (2002). All the truth about NEvAr. *Applied Intelligence, Special Issue on Creative Systems*, *16*(2), 101–119.

Machado, P., Correia, J., & Assunção, F. (2015). Graph-based evolutionary art. In A. Gandomi, A. H. Alavi, & C. Ryan (Eds.), *Handbook of Genetic Programming Applications*. Berlin: Springer.

Machado, P., Correia, J., & Romero, J. (2012a). Expression-based evolution of faces. In *Evolutionary and Biologically Inspired Music, Sound, Art and Design - First International Conference, EvoMUSART 2012, Málaga, Spain, April 11-13, 2012. Proceedings, Theoretical Computer Science and General Issues*

(Vol. 7247, pp. 187–198). Lecture Notes in Computer Science. doi:10.1007/978-3-642-29142-5_17

Machado, P., Correia, J., & Romero, J. (2012b). Improving face detection. In A. Moraglio, S. Silva, K. Krawiec, P. Machado, & C. Cotta (Eds.), *Genetic Programming - 15th European Conference, EuroGP 2012, Málaga, Spain, April 11-13, 2012. Proceedings* (Vol. 7244, pp. 73–84). Lecture Notes in Computer Science. doi:10.1007/978-3-642-29139-5_7

Machado, P., Dias, A., & Cardoso, A. (2002). Learning to colour greyscale images. *The Interdisciplinary Journal of Artificial Intelligence and the Simulation of Behaviour, AISB Journal*, *1*(2), 209–219.

Machado, P., Romero, J., Cardoso, A., & Santos, A. (2005). Partially interactive evolutionary artists. *New Generation Computing, Special Issue on Interactive Evolutionary Computation*, *23*(42), 143–155.

Machado, P., Romero, J., & Manaris, B. (2007). Experiments in computational aesthetics: An iterative approach to stylistic change in evolutionary art. In J. Romero & P. Machado (Eds.), *The Art of Artificial Evolution: A Handbook on Evolutionary Art and Music* (pp. 381–415). doi:10.1007/978-3-540-72877-1_18

Machado, P., Romero, J., Santos, A., Cardoso, A., & Pazos, A. (2007). On the development of evolutionary artificial artists. *Computers & Graphics*, *31*(6), 818–826. doi:10.1016/j.cag.2007.08.010

Machado, P., Vinhas, A., Correia, J., & Ekárt, A. (2015). Evolving ambiguous images. In Q. Yang & M. Wooldridge (Eds.), *Proceedings of the 24th International Joint Conference on Artificial Intelligence, IJCAI 2015, Buenos Aires, Argentina, July 25-31, 2015* (pp. 2473–2479). AAAI Press.

Martins, T., Correia, J., Costa, E., & Machado, P. (2015). Evotype: Evolutionary type design. In C. Johnson, A. Carballal, & J. Correia (Eds.), *Evolutionary and Biologically Inspired Music, Sound, Art and Design - 4th International Conference, EvoMUSART 2015, Copenhagen, Denmark, April 8-10, 2015, Proceedings, Lecture Notes in Computer Science* (Vol. 9027, pp. 136–147). Lecture Notes in Computer Science. doi:10.1007/978-3-319-16498-4_13

McCormack, J. (2007). Facing the future: Evolutionary possibilities for human-machine creativity. In J. Romero & P. Machado (Eds.), *The Art of Artificial Evolution: A Handbook on Evolutionary Art and Music* (pp. 417–451). doi:10.1007/978-3-540-72877-1_19

McCormack, J. (2019). Creative systems: A biological perspective. In T. Veale & F. A. Cardoso (Eds.), *Computational creativity: The philosophy and engineering of autonomously creative systems* (pp. 327–352). Springer.

Nakhmani, A., & Tannenbaum, A. (2013). A new distance measure based on generalized image normalized cross-correlation for robust video tracking and image recognition. *Pattern Recognition Letters*, *34*(3), 315–321.

Neufeld, C., Ross, B., & Ralph, W. (2007). The evolution of artistic filters. In J. Romero & P. Machado (Eds.), *The Art of Artificial Evolution: A Handbook on Evolutionary Art and Music* (pp. 335–356). doi:10.1007/978-3-540-72877-1_16

Nguyen, A. M., Yosinski, J., & Clune, J. (2015). Innovation engines: Automated creativity and improved stochastic optimization via deep learning. In S. Silva & A. I. Esparcia-Alcázar (Eds.), *Proceedings of the Genetic and Evolutionary Computation Conference, GECCO 2015, Madrid, Spain, July 11-15, 2015* (pp. 959–966). doi:10.1145/2739480.2754703

Romero, J., & Machado, P. (Eds.). (2007). *The Art of Artificial Evolution: A Handbook on Evolutionary Art and Music*. Natural Computing Series. doi:10.1007/978-3-540-72877-1

Romero, J., Machado, P., Carballal, A., & Correia, J. (2012). Computing aesthetics with image judgement systems. In J. McCormack & M. d'Inverno (Eds.), *Computers and Creativity* (pp. 295–322). doi:10.1007/978-3-642-31727-9_11

Romero, J., Machado, P., Carballal, A., & Santos, A. (2012). Using complexity estimates in aesthetic image classification. *Journal of Mathematics and the Arts*, *6*(2–3), 125–136. doi:10.1080/17513472.2012.679514

Romero, J., Machado, P., Santos, A., & Cardoso, A. (2003). On the development of critics in evolutionary computation artists. In R. Günther et al. (Eds.), *Applications of Evolutionary Computing, EvoWorkshops 2003: EvoBIO, EvoCOMNET, EvoHOT, EvoIASP, EvoMUSART, EvoSTOC* (Vol. 2611). LNCS, Essex, UK: Springer.

Ross, B. J., Ralph, W., & Hai, Z. (2006). Evolutionary image synthesis using a model of aesthetics. In G. G. Yen, S. M. Lucas, G. Fogel, G. Kendall, R. Salomon, B.-T. Zhang, ... T. P. Runarsson (Eds.), *Proceedings of the 2006 IEEE Congress on Evolutionary Computation* (pp. 1087–1094). IEEE Press.

Rubner, Y., Tomasi, C., & Guibas, L. J. (2000). The earth mover's distance as a metric for image retrieval. *International Journal of Computer Vision (IJCV)*, *40*(2), 99–121.

Saunders, R. (2001). *Curious Design Agents and Artificial Creativity — A Synthetic Approach to the Study of Creative Behaviour* (PhD Thesis, University of Sydney, Department of Architectural and Design Science, Sydney, Australia).

Secretan, J., Beato, N., D'Ambrosio, D. B., Rodriguez, A., Campbell, A., Folsom-Kovarik, J. T., & Stanley, K. O. (2011). Picbreeder: A case study in collaborative evolutionary exploration of design space. *Evolutionary Computation*, *19*(3), 373–403.

Sims, K. (1991). Artificial evolution for computer graphics. *ACM Computer Graphics*, *25*, 319–328.

Tamura, H., Mori, S., & Yamawaki, T. (1978). Textural features corresponding to visual perception. *IEEE Transactions on Systems, Man and Cybernetics*, *8*(6), 460–473. doi:10.1109/TSMC.1978.4309999

Vinhas, A. (2015). *Novelty and Figurative Expression-Based Evolutionary Art* (MSc Dissertation, Department of Informatics Engineering, Faculty of Sciences and Technology, University of Coimbra).

Wang, L., Zhang, Y., & Feng, J. (2005). On the euclidean distance of images. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, *27*(8), 1334–1339.