Endogenously Emergent Information Systems

J. Iivari

1 Introduction

Information systems are growing more complex and autonomous systems of systems. Salvaneschi [70] describes it nicely: “Large information systems are composed of dozens of software applications—programs that typically implement a business process or part of it. Applications may be developed in house or acquired from vendors and possibly adapted. During the evolution the information system grows, integrating more and more applications and changing the existing ones. The evolution is managed by different vendors and development teams working only on parts of the whole system” (pp. 8–9).

IBM has suggested four features of autonomic systems: self-configuring, self-optimizing, self-healing and self-protecting [40]. Nielsen et al. [63] characterize systems of systems with autonomy of the constituent systems, their operational independence but interdependence within the whole, distribution, evolution, dynamic reconfiguration, interoperability and emergence. The focus of this paper lies in “emergence”.

Information systems have been characterized as “emergent” by a number of researchers. The term “emergent” and “emergence” are ambiguous, however, with a number of interpretations and meanings. These concepts have been of considerable interest in Computer Science, but not so much in the IS literature.

So, the purpose of this paper is to analyze and clarify the concepts of “emergence” in the context of information systems and to discuss their implications to IS

A prior version of this paper has been published in the ISD2016 Proceedings (http://aisel.aisnet.org/isd2014/proceedings2016).

J. Iivari (✉)
University of Oulu, Oulu, Finland
e-mail: juhani.iivari@oulu.fi

J. Gołuchowski et al. (eds.), *Complexity in Information Systems Development*,
Lecture Notes in Information Systems and Organisation 22,
DOI 10.1007/978-3-319-52593-8_7

research. The paper pays special attention to endogenous dynamic emergence of information systems, implying that "emergence" is due to the complexity of the system and its operational interaction with its environment. There are three reasons for this focus. Firstly, as noted above, information systems have grown more complex, often being systems of systems [70] or at close to them [32]. This complexity makes them prone to emergent behavior. Secondly, some of this emergent behavior may be undesirable as illustrated for example by the unintended sudden acceleration of cars due to software and the anomalous stock market behaviors [74, 92]. Thirdly, existing IS research has largely omitted this endogenous emergence. The reason for this neglect may be that the IS community has not had a special concept to make the phenomenon explicit.

2 The Concept of Emergence

According to [58] "emergence refers the phenomenon whereby the macroscopic properties of a system arise from the microscopic properties (interactions, relationships, structures and behaviours) of its constituents" (p. 422). It has been widely discussed in biology, psychology, physics, systems theory, philosophy and so on [20, 21, 71, 72]. Quite interestingly, it has also been of considerable interest in Computer Science and in particular in Artificial Intelligence (e.g. [1, 2, 37, 60, 84]) but not so much in the IS literature, [39, 46, 58, 61] as exceptions.

"Emergence" continues to be a contested concept and it is difficult, if not impossible, to provide a definition that would be accepted by all. It is often characterized by phrases such as "the whole is more than the sum of its parts", "much coming from little", "coming into being". So, this paper does not attempt to provide any definite definition, but conceptualizes "emergence" in terms of a number of characteristics shared by "emergent" systems. However, in the case of "emergence" I will focus on the "dynamic emergence" rather than on the "static emergence" [1, 2]. Static emergence can be illustrated by emergent properties such as the number of bedrooms in a house or the durability of a spider web. Dynamic emergent properties change over time. They represent emergent behavior as "coming into being" [34].

Dynamic emergent behavior may be designed and in that way anticipated (= functionality) or non-designed and unanticipated. Normally, the designed emergent behavior is desirable, whereas non-designed, unanticipated behavior may be either desirable or undesirable. Figure 1 introduces the resultant classification of emergent properties, inspired by Ferreira et al. [28].

One way to open the concept "emergence" is to look at characteristics of phenomena that are considered emergent: complexity of the system, its interaction with the environment, learning and adaptation, lack of central control, and unpredictability.

Emergence is often associated with complex systems [37] and especially complex dynamic systems [8]. Bar-Yam [8] notes that a complex system of interdependent parts may exhibit complex emergent behavior. There are also explicit attempts in Computer Science to apply the ideas of complex adaptive systems to

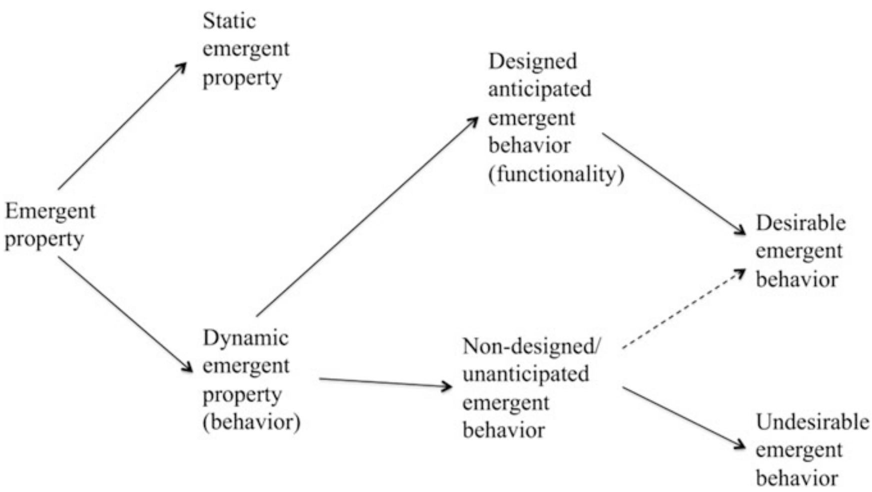

Fig. 1 A classification of emergent properties

develop models of emergent computation that are explicitly based on "emergence" [60]. Emergent behavior or functionality in these computational models cannot be reduced to the behaviors of the agents the system is composed of [29].

Complex dynamic systems are usually open systems. Wegner et al. [90, 91] point out the traditional algorithmic model of computing based on the Turing machine is limited, since Turing machines cannot accept external input while computing.[1] As an alternative [90] proposes "interactive computing", which allows interactions with the environment while computing. If the system simultaneously interacts with numerous environmental objects, it obviously increases the behavioral complexity of the system. So, one can anticipate that not only the internal complexity of the system in terms of the number of its elements and their interdependencies but also the interaction complexity with the environment may produce emergent behavior.

Holland [37] claims that "Any serious study of emergence must confront learning" (p. 53). He illustrates how an adaptive system governed by relatively few and simple rules can exhibit emergent behavior. Although machine learning has a long tradition in Artificial Intelligence, learning information systems have not formed a notable topic in mainstream IS research.

Holland [37] also associates emergence with lack of central control. One can analyze the existence of centralized control in the operational system—whether there is a subsystem that centrally controls the whole system—and existence of centralized control especially when an information system are developed in a

[1]Note that there seems to be some differences of opinion of what Turing had in mind or what Turing Machine as a model of computing implies [17].

distributed way. Referring to [63] one can argue that the trend is towards systems without centralized control.

Predictability has been discussed especially in the context of reducibility of emergent phenomena, i.e. to what extent they (at the system level) can be explained in terms of the lower-level mechanisms (e.g. system components and their interactions). The question has been of interest especially to philosophers [15, 16, 47, 48]. Chalmers [15] distinguishes strong emergence and weak emergence: Strong emergence assumes that an emergent phenomenon arises from the low-level domain, but truths concerning that emergent phenomenon are not deducible even in principle from truths in the lower-level domain. Weak emergence assumes that truths concerning the phenomenon just are unexpected given the principles governing the low-level domain. Kim [47] distinguishes inductive predictability and theoretical predictability of emergent phenomena. He contends that one can inductively predict emergent phenomena: Having observed that an emergent phenomenon E occurs whenever any system has a specific low-level state S, one may predict that a particular system will have the emergent phenomenon E at specific time, if one knows that low-level state of the system will be S at the specific time. Inductive predictability does not imply theoretical predictability so that even full information of the state of the low level domain would not allow prediction of the emergent phenomenon. One objection to the theoretical predictability is that the emergent phenomenon E is not a concept belonging the low-level domain.

In line with [37] the present paper adopts a pragmatic rather than a philosophical position to the predictability of emergent phenomena, focusing especially on information systems. Since the normal functionality of a system is expected emergent behavior of the system, I will focus on the unpredictable emergent behavior that is unexpected, coming as a surprise. Complex software and information systems tend to have such unexpected emergence, since the prediction of their behavior is usually difficult, due to the fact that the state space of the system is very large, the system rarely returns to a state already visited, especially if the system is able to learn and adapt [37, 67].[2]

3 The Concept of Emergence in the Mainstream IS Literature

Since in the ordinary language the word "emerge" may be used as a synonym to words such as "appear" and "rise", it is quite difficult to conduct a bibliographic search on the more "technical" use of the word in the context of information

[2]Predictability is, of course, a matter of degree. Therefore, many qualities of information and software systems such as reliability, maintainability, efficiency can be regarded as emergent properties—they are system-level qualities hard to reduce to the system components and usually not completely predictable.

systems. Based on the authors' familiarity with Information Systems, I tried a forward search [89] in which I used as additional keywords the names of the authors of the early articles on emergence in the context of information systems [36, 53, 57, 65, 78–80], limiting the search to the time after the article was published.

Bibliographic searches, focusing on journal articles, using these lists identified a number of additional articles that refer to "emergence" in the spirit of the previous section (see Table 1). The list is not necessarily exhaustive, but likely representative and indicates continued interest in "emergence" in the context of information systems.

Table 1 summarizes the findings, distinguishing organization, IS development (ISD) process (incl. design and implementation), IS use, and IS artifact as phenomena which may considered "emergent". Table 1 also shows that much of the reviewed literature emphasizes emergence in the context of organizations. Although this literature usually does not explicate the micro phenomenon (agents) that generates the emergence, one can easily construe that an organization is a continuous emergent achievement of its members, stakeholders and other organizations it interacts with.

It is also common to characterize the ISD process as emergent. Although also this stream is not very explicit on the micro phenomena that give rise to emergence, one can imagine that it is an outcome of negotiations between users, managers, designers, vendors and other stakeholders during the ISD process. Furthermore, the ISD design process has become more distributed both organizationally and temporally without (complete) centralized control (see Sect. 4.5). This distribution has made the ISD process and its outcome emergent. Overall, this emergence has resemblance with the ideas of "emergent design" (design as a verb) [14, 25], but contrary to these examples this paper underlines the difficulty of centralized control in this context.

There is also research that considers IS use emergent. Among this stream Nan [61] is conceptualizes the emergence to rise from the interaction between user, technology and task and the interaction between users.

Table 1 also shows that a number of references have recognized the IS artifact as emergent. Yet, it has mostly been quite implicit: the emergent ISD process is assumed to lead to emergent IS artifact so that one can speak about "emergent design" (design as a noun), "emergent requirements", "emergent architecture", "emergent structure", analogously to emergent strategy in [59]. In this case "emergence" is assumed to be an outcome of an emergent development process.

So, the mainstream IS literature cited in Table 1 mostly assumes emergence to be an outcome of exogenous, although, complex design agency. I suggest that "emergence" of IS artifacts may also be more endogenous, inherent to them, resulting from the internal interaction of the subsystems and their interactions with the (dynamic) environment. The purpose of the following section is to have a more detailed look at endogenously emergent behavior of information systems.

Table 1 Primary focus on "emergent" phenomena in the context of information systems

Article	Organization	ISD process	IS use	IS artifact
Markus and Robey [57]	x		x	
Truex and Klein [80]	x	x		x
Lyytinen and Ngwenyama [53]	x	x	x	x
Hirschheim et al. [36]	x	x		x
Iivari and Hirschheim [42]	x	x		x
Orlikowski [65]	x		x	
Ngwenyama [62]	x	x	x	x
Karsten [44]	x	x	x	x
Lycett and Paul [52]	x	x		x
Truex et al. [78]	x	x		x
Truex et al. [79]	x	x		x
Orlikowski [66]		x	x	x
Baskerville and Siponen [7]	x			
Bergman et al. [9]		x		x
Markus et al. [56]	x	x		x
Thompson [76]			x	
Levina [50]		x		x
Luna-Reyes et al. [51]		x		
Allen and Varga [4]	x			
Constantinides and Barrett [18]	x	x	x	x
Corea [19]	x		x	
Curseu [22]	x			
Dreyfus and Iyer [24]				x
Bjørn and Ngwenyama [10]	x			
Patel et al. [68]	x	x		x
Wagner et al. [87]	x	x	x	x
Baker [6]	x	x		
Holmström and Sawyer [38]	x			x
Nan [61]			x	
Essen and Lindblad [26]		x		x

4 Towards Endogenously Emergent Information Systems

4.1 Introduction

Contrary to the IS literature, the issue of endogenously emergent systems has been extensively addressed in biologically inspired Computer Science and Software Engineering (e.g. [11, 29, 40, 55, 63, 83]), in particular. The purpose of this section is to discuss endogenous emergent behavior from the viewpoint of information systems.

This paper interprets an "information system" (IS) to be a computer-based system whose purpose "is to supply its groups of users (…) with information about a set of topics to support their activities" [35].[3] In more morphological terms, an "information system" is a combination of application software and digital information content [86]. According to this interpretation an information system is specific to the organizational (or inter-organizational) context in which it is implemented and that pure software (such as an ERP package) is not an information system.

Following Carvalho [12], we can conceive an information system as a set of interrelated active objects that deal with symbolic objects (information) and whose agents are computers or computer-based devices. Each active object includes a piece of information (or more strictly data embedding or conveying information) and has a number of operations to access data from the environment, to display data, to process data, and to communicate with other active objects of the system. The active objects may either be transient objects or more persistent database objects, storing structured data, electronic documents, websites, knowledge repositories, for example. The granularity of symbolic objects (information content) may vary from simple factual statements to long unstructured documents. In principle, each symbolic object (e.g. each factual statement) may have its own active object.

The definition of an information system is significant, since it naturally has a huge impact on what is endogenous and exogenous. In my view, users and organization belong to the environment of an information system as well as its designers (developers), and an information system is the artifact to be designed and to be used (called "IS artifact" below for brevity).

We recognized internal complexity, dynamic interaction with the environment, learning and adaptation, lack of central control, and unpredictability as typical characteristics of emergent systems. In the following I will argue that modern and especially future IS artifacts share many of these features and are emergent in that sense.

4.2 Internal Complexity

IS artifacts are increasingly complex. The internal complexity of IS artifacts opens the door for emergence as in the case of any complex systems. Furthermore, an IS artifact may be so complex that nobody—a single individual or designer group collectively—understands its totality [82]. As a consequence the system may include or develop unanticipated emergent features. This possibility of emergent features poses considerable reliability, safety and privacy challenges in the case of

[3] I interpret "organization" in a broad meaning here so that in addition to formal organizations it covers more informal organizations such as families and various online communities.

many information and software systems (e.g. [27, 73, 88]). Emergent features as security risks imply that the risk is not because of a local bug, but a result of complex interaction of the component systems, interaction that is extremely difficult to figure out during the design because of the complexity of the system and the question is about run-time interaction.

One should also note that the configuration of an IS artifact in terms of its active objects may be dynamic. In massive, large-scale, wide-area computing networks (such as Internet) and mobile ad hoc networks nodes (active objects) may join and leave the network and connection between nodes may fail [5]. These configurational dynamics naturally increases the complexity of the system and chances of unanticipated emergent behavior.

Such complex IS artifacts may also comprise massive amount of potentially heterogeneous information content (data). Tolk et al. [77] distinguish six levels of interoperability systems. In addition to the technical interoperability (a communication protocol for exchanging data between the systems) and syntactic interoperability (a common structure to exchange data), they identify semantic interoperability (a shared meaning of data, i.e. its information content), pragmatic interoperability (awareness of methods and procedures applied in each subsystem), dynamic interoperability (awareness of the state changes in assumptions and constraints each subsystem implies), and conceptual interoperability (assumptions and constraints of the abstraction of reality).[4] Although the subsystems or components are interoperable at the technical and syntactical level so that the subsystems are able to communicate with each other, it does not assure that the system as a whole exhibits the desired emergent functionality and avoids harmful behavior.

Potentially, this massive information content includes hidden patterns to be discovered. All research on "big data" and data mining rests on this potentiality. These hidden patterns can be considered as static emergent properties, which arise from the individual data and their relationships.

4.3 Interaction with the Environment

IS artifacts have not only grown internally complex, but their interaction with the environment has also become more complex, Internet-of-things as the latest trend. Valckenaers et al. [82] divides problems into one-shot problems and going concerns, claiming that real-life problems are mostly of the latter type. One-shot computational problems can be solved using algorithmic models of computing, but going concerns require interactive computing [90]. Although one should not confuse interactive computing, distributed computing and parallel computing,

[4]These levels are developed keeping simulation models in mind, but illustrate that deliberate and safe integration of systems requires getting acquainted with a huge amount meta-information the systems to be integrated.

obviously interactive computing that is distributed and includes parallelism forms a situation where emergence as the system's unanticipated features and behavior is most likely.

Goldin et al. [30] discuss database-oriented IS artifacts emphasizing that their dynamics can either be algorithmic, sequential interactive, or concurrent interactive —represented by a Turing machine, Sequential (single-stream) Interaction Machine (SIM) or Multi-stream Interaction Machine (MIM) respectively. They note that MIM provides a generic model for the IS dynamics, implying transduction of one or more autonomous input streams from external users into output streams of system feedback, accompanied by an evolving system state.[5] They also claim that the complexity of modeling an IS artifact comes from the complex nature of MIM solution spaces.

When IT becomes more mobile, pervasive and ubiquitous [54], one can expect that also IS artifacts become increasingly context-aware so that they identify and interact possibly with numerous objects in their environments such as users and other systems and objects by sensors and effectors. Context-aware computing has been of considerable research interest during the last twenty years covering for example location-awareness, environment-awareness, artifact-awareness, activity-awareness, participant-awareness, and user-awareness [31, 75].

4.4 Learning and Adaptation

Machine learning has a long tradition in Artificial Intelligence [43, 49], but to my knowledge "learning information systems" is almost a totally neglected area in mainstream IS research. On the other hand, I guess that machine learning has been discussed and applied in the contexts of special areas of IS such "intelligent information systems", "intelligent decision support systems", data mining, document/content management, and knowledge discovery, for example.[6]

The two trends—increased complexity and intensified system-environment interaction—identified above imply that IS artifacts tend to entail huge amount of information content (data) reflecting the dynamic environment. The challenge in this situation is to make sense of the meaning (semantics) of all that heterogeneous information content. It is hard to believe that any a priori defined ontology could solve the issue, but it must take place more inductively in the spirit of emergent semantics [3, 45]. For example, when a higher-level construct (e.g. a pattern in data mining) is identified, this new construct—if found useful by users (domain experts)

[5]Users here can be interpreted to cover not only human users in the external environment of the IS artifact, but also various objects in the environment that generate input streams and/or receive output streams.

[6]One should note that our interest here is not in initial of training of the system to classify documents, for example, but in the automatic learning by the system while in use.

—is made a part of the information content. After that users can refer to them in their interaction with the system and the system itself can structure its information content making use of the aggregated concepts.

4.5 Lack of Centralized Control

There is a clear trend towards interconnected, cooperative IS artifacts composed of independently developed application packages, software components, software agents or web services. When IS artifacts become more like systems of systems with high autonomy (managerial independence) and operational independence [63], centralized operational control of the whole system of systems becomes more unlikely. It is particularly so when the system (of systems) is automatically composed (or dynamically re-configured) or its development has been horizontally distributed without centralized control. By horizontal distribution I mean that the system is development by fairly independent teams (or individual developers) largely simultaneously, while in vertical distribution the development takes place by the same team or different teams sequentially.

Whether distributed or not, the key challenge of coordinating complex software development is the management of dependencies between software components or modules [23, 33]. Although centralized control is considered one of the "best practices" in distributed software development [13], it is difficult in practice because of, the sheer number of dependencies, which tends to explode when the size of software grows, and especially run-time dependencies are difficult or impossible to identify. As a consequence despite modular decomposition, software architecture, Application Programming Interfaces (APIs) and configuration management, software dependencies must also be coordinated by mutual adjustment requiring horizontal communication between developers and teams [13]. In light of all these challenges it is amazing that people have been able to develop such complex systems that have the desired functionality most of the time and at least not fatally harmful emergent behaviors.

4.6 Unpredictability

As mentioned above emergent behavior is to some extent unpredictable. This makes "emergence" challenging in the case of IS artifacts—how to "control" emergence so that exhibits desirable behavior and how to avoid unpredictable harmful behavior [29].

On the other hand, some unpredictability is inherently desirable in the case of IS artifacts, since one can claim that information has value only when it has surprise value.

5 Discussion and Final Comments

Although the IS literature widely refers to emergence, it has mainly focused on "emergent design" (design as a verb) of information systems and implicitly on the "emergent design" (design as a noun) of the resultant the system. It seems to me that there is an opportunity for additional research that studies conceptually and empirically different forms of distributed IS development—such as application-package-based development in the case of multiple suppliers, outsourced IS development with multiple vendors, agile development with multiple teams, free/open source development without centralized control, "re-design in use"—as instances of "emergent design" ("design" both as a verb and as a noun), recognizing that strict centralized control is problematic in these contexts.

The IS literature has largely omitted endogenously emergent behavior of information systems. In view of the fact that information systems are increasingly complex systems of autonomous systems, the question is if IS can afford to omit it and the interaction between such complex systems and different units of adopters (individuals, groups, organizations, markets, communities and societies). A noted above a big research question in the context of endogenous emergence is how "to control" it so that the system exhibits desired system-level behavior and how to avoid harmful system-level behavior, when the design focuses on agents at the micro-level [29]. The possibility of harmful emergent behavior may seriously affect individuals, organizations and society, implying that we—researchers, IS developers, politicians and the general public—should pay attention to risks of such systems [28]. So, it seems to me that there is a clear research opportunity to investigate what the ideas of systems of systems [63] and autonomic computing [40] mean to the above adopting units and stakeholders and to Information Systems as a discipline.

Contrary to Information Systems, Computer Science and Software Engineering have paid considerable attention to endogenous emergence. The question is if Information Systems could make any meaningful contribution to that discourse and open new research perspectives and directions. Overall, I have an impression that the existing literature does not address very explicitly the role of information content as a source of endogenous emergence of information systems.

When reading the literature I also encountered terms "design for emergence" and "design by emergence" [69, 81, 85]. The space does not allow me to discuss these concepts in length, but I would suggest that "design for emergence" refers to design that focuses on designing conditions and constraints to affect exogenous emergence ("emergent design") and endogenous emergence.

"Design by emergence" in my vocabulary is a design process that makes use on endogenous dynamic emergence. If we interpret this emergence as unpredictable, the combination of design and emergence seems a misnormer. However, such emergence may effectively support innovation [64]. In the case of IS artifacts, if the purpose of the system is not just to inform the users but to affect in real-time an ongoing "real" process in its environment, unpredictable emergent behavior may be disastrous. If there is a human being in between interpreting the information, he or she may observe if there is something wrong in the piece of information provided by the system, but not always. However, if the purpose of the system is to explore possibilities, unpredictable endogenous emergent system behavior may be very informative. However, "design by emergence" seems the most promising in computer game design [69, 85] and especially when designing digital fantasizing applications [41]. Design by emergence in their contexts might imply that you cannot play the same game twice or you cannot enter the same fantasy world twice, not only because the constellation of co-participants in the game instance may be different (as in multiplayer games) or the context and physical space of the game may be unique (as in location-based games), but because the game as itself is designed to comprise endogenously emergent functionality.

References

1. Abbot, R.: Emergence explained: abstractions. Complexity **12**(1), 13–26 (2006)
2. Abbot, R.: Putting complex systems to work. Complexity **13**(2), 30–49 (2007)
3. Aberer, K., et al.: Emergent semantics systems. In Semantics of a Networked World. Semantics for Grid Databases. LNCS, vol. 3226, pp. 14–43. Springer (2004)
4. Allen, P.M., Varga, L.: A co-evolutionary complex systems perspective on information systems. J. Inf. Technol. **21**, 229–238 (2006)
5. Babaoglu, O., et al.: Design patterns from biology for distributed computing. ACM Trans. Auton. Adapt. Syst. **1**(1), 26–66 (2006)
6. Baker, E.W.: Why situational method engineering is useful to information systems development. Inf. Syst. J. **21**, 155–174 (2011)
7. Baskerville, R., Siponen, M.: An information security meta-policy for emergent organizations. Logistics Inf. Manag. **15**(5/6), 337–346 (2002)
8. Bar-Yam, Y.: Dynamics of Complex Systems. Addison-Wesley, Reading (1997)
9. Bergman, M., King, J.L., Lyytinen, K.: Large-scale requirements analysis revisited: the need for understanding the political ecology of requirements engineering. Requirements Eng. **7**, 152–172 (2002)
10. Bjørn, P., Ngwenyama, O.: Virtual team collaboration: building shared meaning, resolving breakdowns and creating translucence. Inf. Syst. J. **9**, 227–253 (2009)
11. Brunner, K.A.: What's emergent in emergent computing? In: EMCSR 2002 Conference: 16th European Meeting on Cybernetics and Systems Research, vol. 1, pp. 189–192 (2002)
12. Carvalho, J.A.: Information system? Which one do you mean? In: Falkenberg, E., Lyytinen, K., Verrijn-Stuart, A. (eds.) Information Systems Concepts: An Integrated Discipline Emerging. Kluwer, pp. 259–282 (2000)
13. Cataldo, M., Bass, M., Herbsleb, J.D., Bass, L.: On coordination mechanisms in global software development. In: International Conference on Global Software Engineering (ICGSE 2007). IEEE, pp. 72–80 (2007)
14. Cavallo, D.: Emergent design and learning environments: building on indigenous knowledge. IBM Syst. J. **39**(3&4), 788–821 (2000)
15. Chalmers, D.J.: Strong and weak emergence. In: Clayton, P., Davies, P. (eds.) The Re-emergence of Emergence: The Emergentist Hypothesis from Science to Religion, pp. 1–31. Oxford University Press, Oxford (2006)
16. Clayton, P.: Conceptual foundations of emergence theory. In: Clayton, P., Davies, P. (eds.) The Re-emergence of Emergence: The Emergentist Hypothesis from Science to Religion, pp. 244–254. Oxford University Press, Oxford (2006)

17. Cockshott, P., Michaelson, G.: Are there new models of computation? Reply to Wegner and Eberbach. Comput. J. **50**(2), 232–247 (2007)
18. Constantinides, P., Michael Barrett, M.: Negotiating ICT development and use: the case of a telemedicine system in the healthcare region of Crete. Inf. Organ. **16**, 27–55 (2006)
19. Corea, S.: Mounting effective IT based customer service operations under emergent conditions: deconstructing myth as a basis of understanding. Inf. Organ. **16**, 109–142 (2006)
20. Corning, P.A.: The re-emergence of "emergence": A venerable concept in search of a theory. Complexity **7**(6), 18–30 (2002)
21. Corning, P.A.: The re-emergence of emergence, and the causal role of synergy in emergent evolution. Synthese **185**, 295–317 (2012)
22. Curseu, P.L.: Emergent states in virtual teams: a complex adaptive systems perspective. J. Inf. Technol. **21**, 249–261 (2006)
23. de Souza, C.R.B.: On the Relationships between Software Dependencies and Coordination: Field Studies and Tool Support. Ph.D. dissertation, University of California, Irvine, CA (2005)
24. Dreyfus, D., Iyer, B.: Managing architectural emergence: a conceptual model and simulation. Decis. Support Syst. **46**, 115–127 (2008)
25. Drury, M., Conboy, K., Power, K.: Obstacles to decision making in agile software development teams. J. Syst. Softw. **85**, 1239–1254 (2012)
26. Essen, A., Lindblad, S.: Innovation as emergence in healthcare: unpacking change from within. Soc. Sci. Med. **93**, 203–211 (2013)
27. Felici, M.: Capturing emerging complex interactions: Safety analysis in air traffic management. Reliab. Eng. Syst. Saf. **91**, 1482–1493 (2006)
28. Ferreira, S., Faezipour, M., Corley, H.W.: Defining and addressing the risk of undesirable emergent properties. In: Systems Conference (SysCon), pp. 836–830 (2013)
29. Gleizes, M.-P., Camps, V., George, J.-P., Capera, D.: Engineering systems which generate emergent functionalities. In: Weyns, D., Brueckner, S.A., Demazeau, Y. (eds.) Engineering Environment-Mediated Multi-Agent Systems. LNCS, vol. 5049, pp. 58–75. Springer (2008)
30. Goldin, D., Srinivasa, S., Thalheim, B.: IS = DBS + Interaction: towards principles of information system design. In: Laender, A.H.F., Liddle, S.W., Storey, V.C. (eds.) ER2000 Conference. LNCS, vol. 1920, pp. 140–153. Springer (2000)
31. Gorlenko, L., Merrick, R.: No wires attached: usability challenges in the connected mobile world. IBM Syst. J. **42**(4), 639–651 (2003)
32. Gorod, A., Sauser, B., Boardman, J.: System-of-systems engineering management: a review of modern history and a path forward. IEEE Syst. J. **2**(4), 483–499 (2008)
33. Grinter, R.: Recomposition: coordinating a web of software dependencies. Comput. Support. Coop. Work **12**, 297–327 (2003)
34. Groenewegen, P., Wagnenaar, P.: Managing emergent information systems: Towards understanding how public information systems come into being. Inf. Polity **11**, 135–148 (2006)
35. Gustafsson, M.R., Karlsson, T., Bubenko, J. Jr.: A declarative approach to conceptual information modeling. In: Olle, T.W., Sol, H.G., Verrijn-Stuart, A.A. (eds.) Information Systems Design Methodologies: A Comparative Review, pp. 93–142. North-Holland, Amsterdam (1982)
36. Hirschheim, R., Klein, H.K., Lyytinen, K.: Exploring the intellectual structures of information systems development: a social action theoretic analysis. Account. Organ. Inf. Technol. **6**(1/2), 1–64 (1996)
37. Holland, J.H.: Emergence: From Chaos to Order. Addison-Wesley (1999)
38. Holmström, J., Sawyer, S.: Requirements engineering blinders: exploring information systems developers' black-boxing of the emergent character of requirements. Eur. J. Inf. Syst. **20**(1), 34–47 (2011)
39. Hovorka, D., Germonprez, M.: Perspectives on emergence in information systems research. Commun. Assoc. Inf. Syst. **33**, 353–364 (2013)

40. Huebscher, M.C., McCann, J.A.: A survey of autonomic computing—degrees, models, and applications. ACM Comput. Surv. **40**(3), 7-1–7-28 (2008)
41. Iivari, J.: Paradigmatic analysis of information systems as a design science. Scand. J. Inf. Syst. **19**(2), 39–63 (2007)
42. Iivari, J., Hirschheim, R.: Analyzing information systems development: a comparison and analysis of eight IS development approaches. Inf. Syst. **21**(7), 551–575 (1996)
43. Jordan, M.I., Mitchell, T.M.: Machine learning: trends, perspectives, and prospects. Science **349**(6245), 255–260 (2015)
44. Karsten, H.: Collaboration and collaborative information technologies: a review of the evidence. DATA BASE Adv. Inf. Syst. **30**(2), 44–64 (1999)
45. Kaufmann, M., Wilke, G., Portmann, E., Hinkelmann, K.: Combining bottom-up and top-down generation of interactive knowledge maps for enterprise search. In: Buchmann, R. et al. (eds.) KSEM 2014. LNAI, vol. 8793, pp. 186–197. Springer (2014)
46. Kaufmann, M.A., Portmann, E.: Biomimetics in design-oriented information systems research. In: Donnellan, B., et al. (eds.) At the Vanguard of Design Science: First Impressions and Early Findings from Ongoing Research Research-in-Progress Papers and Poster Presentations from the 10th International Conference, DESRIST 2015, Dublin, Ireland, 20–22 May, pp. 53–60 (2015)
47. Kim, J.: Making sense of emergence. Philos. Stud. **95**, 3–36 (1999)
48. Kim, J.: Emergence: core ideas and issues. Synthese **151**, 547–559 (2006)
49. Langley, P., Simon, H.A.: Applications of machine learning and rule induction. Commun. ACM **38**(11), 55–64 (1995)
50. Levina, N.: Collaborating on multiparty information systems development projects: a collective reflection-in-action view. Inf. Syst. Res. **16**(2), 109–130 (2005)
51. Luna-Reyes, L.F., Zhang, J., Gil-Garcia, J.R., Cresswell, A.M.: Information systems development as emergent socio-technical change: a practice approach. Eur. J. Inf. Syst. **14**, 93–108 (2005)
52. Lycett, P., Paul, R.J.: Information systems development: a perspective on the challenge of evolutionary complexity. Eur. J. Inf. Syst. **8**, 127–135 (1999)
53. Lyytinen, K.J., Ngwenyama, O.K.: What does computer support for cooperative work mean? A structurational analysis of computer supported cooperative work. Account. Organ. Inf. Technol. **2**(1), 19–37 (1992)
54. Lyytinen, K., Yoo, Y.: Special issue: Issues and challenges in ubiquitous computing. Commun. ACM **45**(12), 62–65 (2002)
55. Macías-Escrivá, F.D., Rodolfo Haber, R., Raul del Toro, R., Hernandez, V.: Self-adaptive systems: a survey of current approaches, research challenges and applications. Exp. Syst. Appl. **40**, 7267–7279 (2013)
56. Markus, M.L., Majchrzak, L.A., Gasser, L.: A design theory for systems that support emergent knowledge processes. MIS Q. **26**, 179–212 (2002)
57. Markus, M.L., Robey, D.: Information technology and organizational change: causal structure in theory and research. Manag. Sci. **34**(5), 583–598 (1988)
58. Merali, Y.: Complexity and information systems. In: Mingers, J., Willcocks, L. (eds.) Social Theory and Philosophy of Information Systems, pp. 407–446. Wiley, London (2004)
59. Minzberg, H.: The Rise and Fall of Strategic Planning. Prentice-Hall, Hemel Hempstead, Hertfordshire, UK (1994)
60. Müller-Schloer, C., Sick, B.: Emergence in organic computing systems: discussion of a controversial concept. In: Yang, L.T., et al. (eds.) ATC 2006. LNCS, vol. 4158, pp. 1–16. Springer (2006)
61. Nan, N.: Capturing bottom-up information technology use processes: a complex adaptive systems model. MIS Q. **35**(2), 505–532 (2011)
62. Ngwenyama, O.K.: Groupware, social action and organizational emergence: on the process dynamics of computer mediated distributed work. Account. Organ. Inf. Technol. **8**, 127–146 (1998)

63. Nielsen, C.B., Larsen, P.G., Fitzgerald, J., Woodcock, J., Peleska, J.: Systems of systems engineering: basic concepts, model-based techniques, and research directions. ACM Comput. Surv. 48(2), 18:1–18:41 (2015)
64. Nijs, D.E.L.W.: The complexity-inspired design approach of imagineering. World Futures **72** (1–2), 8–25 (2015)
65. Orlikowski, W.J.: Improvising organizational transformation over time: a situated change perspective. Inf. Syst. Res. **7**(1), 63–92 (1996)
66. Orlikowski, W.J.: Using technology and constituting structures: a practice lens for studying technology in organizations. Organ. Sci. **11**(4), 404–428 (2000)
67. Parnas, D.L.: Software aspects of strategic defense systems. Commun. ACM **28**(12), 1326–1335 (1985)
68. Patel, N.V., Eldabi, T., Khan, T.M.: Theory of deferred action agent-based simulation model for designing complex adaptive systems. J. Enterp. Inf. Manag. **23**(4), 521–537 (2010)
69. Reid, J., Hull, R., Clayton, B., Melamed, T., Stento, P.: A research methodology for evaluating location aware experiences. Pers. Ubiquit. Comput. **15**, 53–60 (2011)
70. Salvaneschi, P.: Modeling of information systems as systems of systems through DSM. In: SESoS'16, Austin, TX, USA, pp. 8–11 (2016)
71. Stephan, A.: Varieties of emergentism. Evol. Cogn. **5**(1), 49–59 (1999)
72. Symons, J.: Computational models of emergent properties. Mind. Mach. **18**, 475–491 (2008)
73. Taleb-Bendiab, A., England, D., Randles, M., Miseldine, P., Murphy, K.: A principled approach to the design of healthcare systems: Autonomy vs. governance. Reliab. Eng. Syst. Saf. **91**, 1578–1585 (2006)
74. Tanriverdi, H., Rai, A., Venkatraman, N.: Reframing the dominant quests of information systems strategy research for complex adaptive business systems. Inf. Syst. Res. **21**(4), 822–834 (2010)
75. Tarasewich, P.: Designing mobile commerce applications. Commun. ACM **46**(12), 57–60 (2003)
76. Thompson, M.P.A.: Cultivating meaning: interpretive fine-tuning of a South African health information system. Inf. Organ. **12**, 183–211 (2002)
77. Tolk, A., Diallo, S.Y., Turnitsa, C.D.: Applying the levels of conceptual interoperability model in support of integratability, interoperability, and composability for system-of-systems engineering. Syst. Cybern. Inform. **5**(5), 65–74 (2007)
78. Truex, D., Baskerville, R., Klein, H.: Growing systems in emergent organizations. Commun. ACM **42**(8), 117–123 (1999)
79. Truex, D.P., Baskerville, R., Travis, J.: Amethodological systems development: the deferred meaning of systems development methods. Account. Manag. Inf. Technol. **10**, 53–79 (2001)
80. Truex III, D.P., Klein, H.K.: A rejection of structures as a basis for information systems development. In: Stamper, R.K., Kerola, P., Lee, R., Lyytinen, K. (eds.) Collaborative Work, Social Communications and Information Systems, pp. 213–235. Elsevier (North-Holland), Amsterdam (1991)
81. Ulieru, M., Doursat, R.: Emergent engineering: a radical paradigm shift. J. Auton. Adapt. Commun. Syst. **4**(1), 39–60 (2011)
82. Valckenaers, P., Van Brussel, H., Hadeli, Bochmann, O., Saint Germain, B., Zamfirescu, C.: On the design of emergent systems: an investigation of integration and interoperability issues. Eng. Appl. Artif. Intell. **16**, 377–393 (2003)
83. van Steen, M., Pierre, G., Voulgaris, S.: Challenges in very large distributed systems. J. Internet Serv. Appl. **3**, 59–66 (2012)
84. Varenne, F., Chaigneau, P., Petitot, J., Doursat, R.: Programming the emergence in morphogenetically architected complex systems. Acta. Biotheor. **63**, 295–308 (2015)
85. Vogiazou, Y., Raijmakers, B., Geelhoed, E., Reid, J., Eisenstadt, M.: Design for emergence: experiments with a mixed reality urban playground game. Pers. Ubiquit. Comput. **11**, 45–58 (2007)
86. Väyrynen, K., Iivari, J.: The competitive potential of IT applications—an analytical-argumentative evaluation. In: ICIS 2016 (2015)

87. Wagner, E.L., Newell, S., Piccoli, G.: Understanding project survival in an ES environment: a sociomaterial practice perspective. J. AIS **11**(5), 278–297 (2010)
88. Wears, R.L., Cook, R.I., Perry, S.J.: Automation, interaction, complexity, and failure: a case study. Reliab. Eng. Syst. Saf. **19**, 1494–1501 (2006)
89. Webster, J., Watson R.T.: Analyzing the past to prepare for the future: writing a literature review. MIS Q. **26**(2), xiii–xxiii (2002)
90. Wegner, P.: Why interaction is more powerful than algorithms? Commun. ACM **40**(5), 80–91 (1997)
91. Wegner, P., Eberbach, E.: New models of computing. Comput. J. **47**(1), 4–9 (2004)
92. Wolf, M.: Embedded software in crisis. Computer **49**(1), 88–90 (2016)