World Wide Web (2010) 13:33–59
DOI 10.1007/s11280-009-0078-0

Human Intelligence in the Process of Semantic Content Creation

Katharina Siorpaes · Elena Simperl

Received: 27 March 2009 / Revised: 16 August 2009 /
Accepted: 5 November 2009 / Published online: 2 December 2009

Abstract Despite significant progress over the last years the large-scale adoption of semantic technologies is still to come. One of the reasons for this state of affairs is assumed to be the lack of useful semantic content, a prerequisite for almost every IT system or application using semantics. Through its very nature, this content can not be created fully automatically, but requires, to a certain degree, human contribution. The interest of Internet users in semantics, and in particular in creating semantic content, is, however, low. This is understandable if we think of several characteristics exposed by many of the most prominent semantic technologies, and the applications thereof. One of these characteristics is the high barrier of entry imposed. Interacting with semantic technologies today requires specific skills and expertise on subjects which are not part of the mainstream IT knowledge portfolio. A second characteristic are the incentives that are largely missing in the design of most semantic applications. The benefits of using machine-understandable content are in most applications fully decoupled from the effort of creating and maintaining this content. In other words, users do not have a motivation to contribute to the process. Initiatives in the areas of the Social Semantic Web acknowledged this problem, and identified mechanisms to motivate users to dedicate more of their time and resources to participate in the semantic content creation process. Still, even if incentives are theoretically in place, available human labor is limited and must only be used for those tasks that are heavily dependent on human intervention, and cannot be reliably automated. In this article, we concentrate on this step in between. As a first contribution, we analyze the process of semantic content creation in order to identify those tasks that are inherently human-driven. When building semantic applications involving these specific tasks, one has to install incentive schemes that are likely to encourage users to perform exactly these tasks that crucially rely on manual input. As a second

K. Siorpaes (✉) · E. Simperl
Semantic Technology Institute (STI) Innsbruck, University Innsbruck, Innsbruck, Austria
e-mail: katharina.siorpaes@sti2.at

E. Simperl
e-mail: elena.simperl@sti2.at

contribution of the article, we propose incentives or incentive-driven tools that can be used to increase user interest in semantic content creation tasks. We hope that our findings will be adopted as recommendations for establishing a fundamentally new form of design of semantic applications by the semantic technologies community.

Keywords semantic technologies · semantic content · semantic content creation · ontologies · annotation · alignment · human intelligence · human factor

1 Introduction

> *"The original Scientific American article on the Semantic Web appeared in 2001. It described the evolution of a Web that consisted largely of documents for humans to read to one that included data and information for computers to manipulate. The Semantic Web is a Web of actionable information — information derived from data through a semantic theory for interpreting the symbols. The semantic theory provides an account of "meaning" in which the logical connection of terms establishes interoperability between systems. [...] This simple idea, however, remains largely unrealized."*
>
> *(Nigel Shadbolt, Wendy Hall, Tim Berners-Lee (2007). The Semantic Web Revisited. IEEE Intelligent Systems.)*

One reason for this state of affairs is the lack of critical mass of semantic content: real-world, properly maintained ontologies, as well as useful RDF data would leverage a wide adoption of semantic technologies, by facilitating the development of semantic applications in various domains.[1] In the past decade, a significant share of the efforts around the creation and management of semantic content has targeted a complete, or at least partial, automation of these processes, with the aim to enable efficiency and growth. Despite impressive progress in terms of models, methods and software, experiences in numerous case studies across vertical domains show that specific aspects of creating and managing semantic content still rely on a considerable amount of manual effort. Whilst gradual improvements of the limitations of existing tools are to be expected, full automation in this context is, we argue, not feasible, due to the very nature of the tasks related to building and maintaining ontologies, and semantically annotating documents, images and videos based on these ontologies.

Furthermore, while human contribution is indispensable at many points in the process of semantic content creation, the question of why and how the human user should contribute remains to be investigated in depth. The entrance barrier for many semantic technologies and applications is high, as they require expertise in knowledge engineering, or even logics. Recently, the Social Semantic Web has gained momentum triggered by the success of the Web 2.0 phenomenon yielding tools that allow collaboration and participation by lay users [9, 53, 65, 67, 79, 81]. Most of the work undertaken in this area builds on the fact that Web 2.0 applications enjoy amazing popularity; people are willing to spend time adding tags and extending tag sets, while the "heavyweight" Semantic Web lacks sufficient user involvement almost everywhere. Web 2.0 demonstrates not only the importance of usability, but

[1] We use "data" and "content" as synonyms in this work.

also the necessity of proper incentives: users need a motivation why they should dedicate their time to a task, in our case to creating and maintaining OWL or RDFS ontologies and RDF data sets. Stiil, even if incentives are theoretically in place, the available human labor is limited. Therefore, one has to make sure that this valuable resource is only used for those tasks that cannot be automated and that an optimal combination of human and computational intelligence is achieved.

The first contribution of this paper is a comprehensive analysis of the process of semantic content creation with the purpose of identifying those tasks that crucially depend on human intervention, or that have to be inherently human-driven to yield success. We first provide an overview of the state of the art in the area of semantic content creation. We argue that most of the approaches significantly rely on human intervention, and that the barrier of entry of the technology is still too high for most Internet users. Having gained a thorough understanding of the models, methods and tools for semantic content creation, we evaluate the human contribution that is required in order to complete specific tasks in the process in a survey covering 46 tools and 21 methodologies. We investigate each task of semantic content creation and indicate whether it is manual, semi-automatic or automatic. When building semantic applications heavily involving manual tasks, one has to install incentive schemes that are likely to encourage users to perform exactly these tasks that crucially rely on user input. As a second contribution of the article, we propose incentives or incentive-driven tools that can be used to increase user interest in semantic content creation tasks. We hope that our findings will be adopted as recommendations for establishing a fundamentally new form of design of semantic applications by the semantic technologies community.

2 The process of semantic content creation

At the heart of semantic content creation are ontologies: shared specifications of conceptualizations of a domain [36]. Ontology engineering is a challenging process, which encompasses a broad range of activities and tasks, as elaborated in the following cf. by [35].

Ontology management refers to scheduling, controlling and quality assurance. Scheduling is about coordinating and managing an ontology development project, including resource and time management. Controlling ensures that the scheduled tasks are accomplished as planned. Finally, quality assurance evaluates the quality of the outcomes of each activity, most notably of the implemented ontology.

Ontology development is split up in three phases: pre-development, development, and post-development. An environment study investigates the intended usage and goal of the ontology. The feasibility study ensures that the ontology can actually be built within the time and resources assigned to the project. Once the pre-development phase has been completed, the specification activity is carried out and the intended usage, users, and the scope of the ontology are defined. During conceptualization the domain knowledge is structured in meaningful models. During formalization the conceptual model is formalized in order to prepare the implementation during which the model is serialized in an ontology language. In the post-development phase, the ontology is updated and maintained as required. Post-development also includes re-use by other ontologies or applications.

Support covers a wide range of crucial activities that cover many areas of semantic content creation and maintenance. There is not a pre-defined order in which these activities should be undertaken; they can or should be in parallel with the development activities. Typical support activities include knowledge acquisition, evaluation, integration, documentation, merging, configuration management, and alignment. Knowledge acquisition is usually associated with the specification of the ontology. The activity of (semi-)automatically creating and/or populating ontologies is referred to as ontology learning. During evaluation a detailed analysis of the ontology is carried out. Integrations needs to be carried out when other ontologies are re-used for building a new ontology. Merging is a related support activity that produces a new ontology by combining existing ontologies. Ontology alignment establishes relations among related or complementary ontologies. The documentation activity provides a detailed description of the ontology and of the main activities and tasks of the development process. Finally, configuration management tracks versions of the ontology and its documentation, as well as all necessary files.

For the analysis performed in this article, we focus on development and support activities because they are specific to the scope of semantic content creation. In the following, we elaborate on these activities. We also survey some of the most important approaches in the area of Social Semantic Web.

2.1 Ontology development

Developing ontologies requires domain expertise and the ability to capture domain knowledge in a clean conceptual model. An ontology describes the things that are important in a specific domain of interest, their properties, and the way they are interrelated. It defines a common vocabulary and the meaning of this vocabulary. In the last decade, several ontology development methodologies have been proposed. Many suggest to start with the specification of the scope the ontology should cover and the requirements it should fulfil. This is often complemented by the informal and formal specification of competency questions. Based on that, relevant terms in the domain are then collected. Widely accepted ontology representation formalisms such as OWL or RDF(S) use classes, properties, instances and relations as ontological primitives to describe domains. Ontologies can vary in their degree of formality, ranging from highly informal (defined in natural language), to semi-informal (language is restricted), semi-formal (formally defined language), to rigorously formal [35]. As such, and despite existing semi-automated approaches to ontology learning, ontology development remains a costly and time-consuming human-driven process that can be performed in a centralized (within a pre-defined team of engineers and domain experts) or a decentralized fashion (within a potentially open community of stakeholders, domain experts and users).

2.2 Semantic annotation

Ontology population is also referred to as "semantic annotation". There are a wide range of approaches that carry out semi-automatic annotation of texts: most of them make use of natural language processing and information extraction techniques. Even though they require training, a large share of the work can be automated [64, 75]. The situation is slightly different with the annotation of multimedia content: approaches for the annotation of media, no matter if manual, semi-automatic or automatic, aim at closing the so-called "semantic gap", which is a term coined to describe the discrepancy between low-level technical features of multimedia, which can be automatically processed to a great extent, and the high-level, meaning-bearing features a user is typically interested in and refers to when searching for content. Recent research in the area of semantic multimedia retrieval attempts to derive meaning from low-level features, or other available basic meta-data, automatically. Despite continuous advances, multimedia management remains a challenging problem, because the meaning of the content is not localized in the media that is being analyzed [6]. Two streams of research have emerged as a result of this state of affairs: the first one aim to provide rich annotations created by humans as training data for automated analysis, thus depending on human contribution. The second one relies purely on the analysis of the raw multimedia content, performing well in settings in which the relevant concepts can easily be recognized. Richer semantics, capturing implicit features and meaning derived by humans, cannot be extracted in this manner. Popular examples on the Web show that to date there are many service-based platforms that make use of the users' knowledge to understand the meaning of multimedia content: on the photo sharing platform Flickr,[2] users annotate their pictures using tags; something similar is done on YouTube[3] with annotations.

2.3 Ontology support

Support activities accompany the development of ontologies. One prominent example thereof is ontology learning, which can understood as semi-automatic or automatic support when building an ontology [52]. This means that knowledge is acquired in the form of ontological data from various information sources [34]. Most approaches in ontology learning require human input at run time. Another important support activity is the alignment of heterogeneous ontologies. Many of the existing ontology engineering environments provide means for the manual definition of mappings between ontologies. In addition, there is a wide range of algorithms that provide automatic support [24–26, 58, 59], whilst it is generally accepted that the question of which ontological primitives match cannot (yet) be done fully automatically [25, 27]. A last support activity worthwhile being mentioned is ontology evaluation. So far, no fully automatic approaches have emerged, whereas semi-automatic approaches are rare [8].

2.4 Social semantic web

The fields referred to as the "Social Semantic Web" was triggered by the advent of Web 2.0. Bringing a social novelty, rather than a technical one, Web 2.0 rapidly became a worldwide phenomenon which facilitated people from all over the world to connect and collaborate on various topics. Web 2.0 stands for a new era of

[2] http://www.flickr.com

[3] http://www.youtube.com

usability and user participation. The newly emerged tools allow more interaction and exchange. Wiki-based systems, such as Wikipedia,[4] and social tagging systems such as del.icio.us[5] or Flickr are popular examples. The vocabularies jointly created in the social tagging process are used for deriving ontologies [55]. Web 2.0 and the related technologies and applications built upon them provide incentives to their users, and lead to rapidly growing amounts of content.

Inspired by this phenomenon, many collaborative and community-driven approaches to semantic content creation have been proposed. These approaches share the aim of handing back the control over the emergence of an ontology or the creation of semantic annotation to their user base. They also ground on the assumption that human contribution is required for many tasks in the process of semantic content creation. A good example for these tools are semantic wikis. Semantic wikis were a first attempt to exploit Web 2.0 principles and technologies to facilitate broad user participation in the process of creating semantic data [11, 65, 70, 79]. Semantics are also applied in blogging to allow for enhanced information retrieval and management techniques. There are several approaches bringing together Web 2.0 and ontology engineering. Braun and colleagues present an ontology maturing process based on Web 2.0 concepts consisting of four steps: emergence of ideas, consolidation in communities, formalization, and axiomatization [9]. They regard the evolution of an ontology as the community-driven process of maturing from tags to more formal knowledge structures. myOntology is a methodology and platform for community-driven creation and maintenance of lightweight domain ontologies based on a wiki [69]. OntoGame takes a different path to stimulating user participation, by designing a series of games that hides tasks related to semantic content creation behind entertaining multi-player games [67].

We now turn to an analysis of the activities mentioned so far from the point of view of the extent to which they rely on human contributions in order to yield feasible results.

3 Human intelligence in semantic content creation

Where as many Web 2.0-inspired approaches to semantic content creation acknowledge the importance of incentives to encourage user participation, the available human labor remains finite. Complementarily, there are aspects of this process in which automatic approaches has proved to perform reliably, and faster and more accurate than an individual user, or or a community of user, ever could. The question that this section attempts to answer is thus about the best way to combine human and computational intelligence in order to optimize the cost/performance ratio of producing semantic data. To do so, we analyze the activities surveyed in the previous section in the light of their potential to be feasibly automated and identify those tasks for which human input is indispensable. Section 4 then proposes incentive schemes for these tasks as a means to motivate users to perform these tasks.

[4] http://www.wikipedia.org

[5] http://del.icio.us

3.1 Ontology development

We analyzed all methodologies for ontology engineering published in the ontology engineering literature of the last two decades. In this survey, we considered traditional ontology engineering approaches, such as the Cyc method [49], Uschold and King's method [76], Grueninger and Fox' method [37], KACTUS [5], Methontology [31], the SENSUS method [35], the OTK methodology [73, 74], and the Ontology 101 methodology[57]. Additionally, we looked into collaborative methodologies such as DILIGENT [82], HCOME [47], Dogma [18, 42, 72], Ontology Maturing [9], Holsapple and Joshi's approach [41], and UPON [56]. For all methodologies and methods surveyed we relied on the publications describing the corresponding approaches, but also descriptions of case studies in which these approaches have been applied, in particular with respect to the automatic tools available or recommended to assist the ontology engineering team.

Based on this survey, we extracted the most common tasks within the semantic content creation process, and assigned a judgment on their nature, that is, if they are human-driven or, in contrast, automatable. For each of the tasks, we provided the rationales behind our classification. As a reference for the task list presented below we tool the Ontology 101 methodology [57], which can be seen as the common denominator of most of the ontology engineering approaches proposed so far. We did not consider activities related to project management because they are not specific to ontology engineering, or semantic content creation. The results are summarized in Table 1.

Description of domain and scope This task results in a thorough description of what the ontology is expected to cover. It outlines the domain, scope and focus of the ontology and lists the intended uses and users. This activity can also involve defining requirements that the ontology must or should fulfill. In collaborative ontology engineering, this task aims at establishing consensus among the development team. As in many other engineering disciplines, requirements analysis and gathering is mostly a manual process.

Competency questions This task can be seen as a continuation of the previous one: competency questions are a means to further specify the domain and scope of the ontology and to evaluate the ontology in a later stage. They capture the queries that the ontology is expected to answer based on the identified usage scenarios. Competency questions can be either formal or information. Collecting competency questions that reflect the ontology engineering requirements is a challenging conceptual task. This also applies to the formalization of such questions in a language with a machine-understandable semantics. Therefore, we argue that in both cases human input is crucially required.

Re-use of existing ontologies Based on the state of the art, automation support for the re-use of existing ontologies seems feasible when finding potential re-use candidates. The evaluation and selection of such candidates is a more challenging task, where automation is hardly possible. Additionally, the customization of the relevant ontologies to the characteristics of the new application setting is not fully automatable, though tools for extracting ontology fragments, translating to different knowledge representation formalisms, or even ontology alignment are helpful.

Table 1 Classification of tasks in ontology development.

Task	Nature	Comment
Description of domain and scope	Human	The domain and scope (requirements, motivating scenarios) of the ontology must be defined by one or more human actors.
Competency questions	Human	The formulation of informal or formal competency questions is complex and requires a deep understanding of the whole project, which can be hardly done automatically.
Re-use of existing ontologies	Human, computational support partly possible	Discovery of suitable ontologies can be performed automatically. The selection of re-usable ontologies and the actual re-use rely on human input.
Collection of relevant terms	Human, computational support partly possible	Given an appropriate corpus that can be used for term extraction, automation is possible. The final selection of relevant terms must be done by a human actor.
Typing of terms	Human, computational support partly possible	Deciding whether a term falls in a particular category can be supported automatically to a very limited extent. It is largely human-driven.
Building a hierarchy	Human	Building a hierarchy requires human input at least for validation purposes.
Define properties	Human, computational support partly possible	The definition of properties can be supported by automatic methods, however to a less extent than the definition of the class hierarchy.
Define axioms	Human	Extending an ontology with axioms requires a human author.
Create ontological instances	Human	The distinction between classes and instances is very specific and requires the human in the loop.
Documentation	Human	Ontology documentation is heavily human-driven, as it refers to decisions made during the engineering process by the participants.

Collection of relevant terms For the collection of relevant terms Uschold and King propose three different strategies: bottom-up, top-down and middle-out. Bottom-up means to start at a finer-grained level of specificity and to generalize to broader terms. Top-down is the opposite approach in which abstract concepts are identified first and are then specialized into more detailed, narrower concepts. Finally, middle-out is a combination of the previous, in which the most relevant terms are selected first, regardless their level of abstraction, in order to specialize or generalize as required. The outcome of this task is the vocabulary used by the ontology.

Automatic support seems possible when it comes to going through a set of relevant documents and extracting terms that occur often. However, the question of which terms belong to the ontology and which belong to the knowledge base (i.e., the distinction between classes and instances which is required in some ontology engineering methodologies or representation formalisms) has to be answered by a human. Natural language processing tools could suggest a list of relevant terms to a human user, who in turn selects the terms. Summing up, the task can be achieved by machine-supported manual work.

Typing of terms This task answers the question whether a term should be modeled as a concept, an individual, a relationship, or an attribute. The task is highly human-driven because the decisions to be made require a deep understanding of conceptual modeling as well as the formal semantics of the knowledge representation language at hand. Automation would be possible only to the extent that certain assumptions are taken (e.g., all nouns are supposed to be modeled as concepts). Nevertheless, these assumptions need to be validated on a case-by-case basis by a human actor.

Building a hierarchy The goal of this task is to define the class hierarchy. Automation support is possible. For instance, in a top-down approach specialization/generalization relationships can be assigned automatically provided an appropriate corpus of domain knowledge. However, human validation is still needed, as one has to ensure that relationships of a more domain-specific nature hold within the class hierarchy.

Define properties This task includes the definition of cross-taxonomical relations, as well as attributes further describing a concept. Automatic support seems feasible in terms of extracting adjectives, verbs and verbal phrases using natural language processing techniques, as potential properties' candidates. Still, a significant amount of manual work is required to select a suitable corpus of knowledge and to validate the results.

Define axioms Defining axioms involves specifying precise logics-based rules, such as cardinality constraints, disjointness, etc. that apply to concepts. Approaches for automatically specifying such axioms exist. However, they are limited to very specific application areas, and require substantial training and/or extensive validation.

Create instances The creation of instances is also referred to as semantic annotation or ontology population; we investigate it in detail in the next section. Relevant for the context of ontology development is the definition of so-called "fixed" or "ontological" instances. An example would be the need of the instance of city "New York" when specifying the concept "New Yorker"—as a New Yorker obviously always lives in New York. The distinction between classes and instances is very specific to the application setting, and we are not aware of any approaches aiming at automatizing this task.

Documentation The documentation of an ontology is an essential component of ontology engineering to facilitate maintenance and re-use, and to ensure a smooth operation of a collaborative process. Documentation remains human-driven, especially when it comes to recording modeling decisions and their rationales. It can be supported with lexical techniques and by adding knowledge retrieved from the Web.

Summing up, conceptual modeling is to a considerable extent human-driven. Few other tasks show a similar degree of dependence on the human user as this one.

In some cases, the task can be assisted with automatic techniques. Examples include discovering candidate ontologies for re-use, machine learning and information extraction techniques for collecting relevant terms or for turning semi-structured knowledge corpora such as folksonomies [71, 77] into ontologies.

3.2 Semantic annotation

For the analysis of semantic annotation tasks we surveyed a series of tools that partly allow automation. We neglected fully manual ones, as they are per definition dependent on human input. In our analysis we distinguished among the type of content annotated, i.e., text, multimedia, and Web services.

Semantic annotation of text Natural language processing techniques are extensively used to automate text annotation. Most of the available tools require training sets to carry out the annotation autonomously. The results of our analysis are depicted in Table 2.

AeroDaml is a pattern-based approach based on the markup language DAML [46]. The system extracts information and maps it to RDF triples. The underlying ontologies for achieving this task are the WordNet synset hierarchy [29] and the AeroText knowledge base. The tool can map terms that occur in the text to instances of classes or instances of properties. The tool comes in two flavors: as a Web version with a default generic ontology that contains general-interest knowledge, and in a client-server version which supports customized ontologies.

Armadillo [22] is a pattern-based approach to annotation and makes use of the Amilcare information extraction system [16]. It is especially suitable for highly structured Web pages. The tool starts from a seed pattern and does not require human input initially. The patterns for entity recognition have to be added manually though.

The knowledge and information management platform (KIM) [63] consists of the following components: an ontology and knowledge base as well as an indexing and retrieval server. RDF data is stored in an RDF repository [45], whilst search is

Table 2 Classification of approaches approaches to semantically annotate text.

Approach	Nature	Comment
AeroDaml	Automatic	Pre-defined ontology
	Semi-automatic	Adapted ontology
Armadillo	Automatic	Pre-defined patterns
	Semi-automatic	Adapted patterns
KIM	Automatic	Limited focus
Magpie	Automatic	Pre-defined ontology
	Semi-automatic	Adapted ontology
Melita	Manual	Without training
	Semi-automatic	With training
MnM	Manual	Without training
	Semi-automatic	With training
Pankow	Automatic	Limited focus
S-Cream	Manual	Without training
	Semi-automatic	With training
SemTag	Automatic	Limited focus

performed using the LUCENE system.[6] KIM is based on an underlying ontology (KIMO or PROTON) that holds the knowledge required to semantically annotate documents, and on GATE [7] to perform information extraction.

Magpie [23] is a suite of tools that supports the fully automatic annotation of Web pages, by mapping entities found in its internal knowledge base against those identified on Web pages. The quality of the results depends on the background ontology, which has to be manually modeled and populated.

Melita [17] works with Amilcare and can learn from user input by taking annotated content and generalizing this content in order to make new annotations. Amilcare [16] is not an annotation tool per se, but an information extraction system that was adapted for several semantic annotation tools. In the beginning, the system requires the user to fully annotate documents to bootstrap the learning process. Once it is able to make suggestions to the user, it learns from the user feedback. The system gradually increases the number of suggestions. Their quality improves so that the system can do a large part of the annotations on its own. Furthermore, Melita provides possibilities for rule writing, i.e., advanced users can define rules for automatic annotation.

MnM [78] is a tool for semi-automatic annotation based on the Amilcare system. It uses machine learning techniques and requires a training data set. Provided this training data, the system gradually takes over the annotation process. The classical usage scenario MnM was designed for is the following: while browsing the Web, the user manually annotates selected Web pages in the MnM Web browser. While doing so, the system learns annotation rules, which are then tested against user feedback. The better the system does, the less user input is required.

The PANKOW algorithm [15] is a pattern-based approach to semantic annotation that makes use of the redundant nature of information on the Web. Based on an ontology, the system constructs patterns and combines entities into hypotheses that are validated manually.

S-Cream [39] is another approach to semi-automatic annotation that combines two tools: Ont-O-Mat, a manual annotation editor implementing the CREAM framework, and the previously mentioned Amilcare. S-Cream can be trained for different domains provided the appropriate training data and proposes a set of heuristics for post-processing and mapping of information extraction results to an ontology. Ont-O-Mat provides ways to access ontologies specified in a markup format.

The SemTag system [20] is based on the TAP ontology.[7] In a first phase, the system annotates all occurrences of instances of the ontology. In a second phase, it disambiguates the elements and assigns the correct ontological classes by analyzing context. The TAP ontology is similar to the ontology used in KIM.

The nature of semantic annotation of text can not be easily defined, because it requires training to carry out the annotation process autonomously. As such we argue that this task requires substantial human contribution, also if we take into account that training data sets need to be maintained as the domain of the ontology evolves.

[6] http://jakarta.apache.org/lucene/

[7] http://tap.stanford.edu/tap/papers.html

Table 3 Classification of approaches to semantically annotate multimedia.

Approach	Type	Nature	Comment
AktiveMedia	Image	Semi-automatic	Low-level semantics
Caliph	Image	Automatic	Low-level semantics
SWAD	Image	Automatic	Low- and high-level semantics
MPEG-7 SCDSExtractor	Audio	Automatic	Speech
Transcriber	Audio	Automatic	Speech
4M	Video	Automatic	Low-level semantics
		Semi-automatic	Machine learning techniques
M-OntoMat-Annotizer	Video	Automatic	Low-level semantics
		Semi-automatic	Machine learning techniques

Semantic annotation of multimedia Understanding content of images or videos remains a human-driven task: probably the best example supporting this statement are CAPTCHAs—automatically generated images that can be easily understood by humans, but are very challenging for computers. Just as in the previous section, we considered solely those tools that aim at automation support. Our findings are summarized in Table 3.

The SA methodology [10] is the only dedicated methodology for the annotation of multimedia; however, it is in an early stage and does not describe its stage, activities and tasks in depth. Aktive Media [13], an image annotation tool, extracts low-level semantics in a semi-automatic fashion by making suggestions to the user. Caliph [51] handles low-level semantics of images fully automatically. SWAD[8] also extracts low-level semantics to RDF and adds object annotations in a semi-automatic manner. The MPEG-7 SpokenContent Description Scheme Extractor[9] automatically recognizes speech, on which one can apply text-related annotation methods. The same applies for Transcriber [4]. At the time this article was written, only an infrastructure outline for the 4M system [2] could be found: they aim to automatically extract low-level features of videos and to semi-automatically annotate content using machine learning techniques. Similarly, the M-OntoMat-Annotizer [61] allows automatic extraction of low-level semantics and semi-automatic content annotation based on training for a specific domain, based on user-provided ontologies.

The few tools available cover the extraction of low-level semantics from multimedia. This task can be done automatically, whilst not resolving the real challenge of multimedia annotation, which is related to higher-level features referring to the subject of the content. This is feasible to a limited extent for particular domains or types of media.

Semantic annotation of web services Semantic Web Services can be defined as self-contained, self-describing, semantically marked-up software resources that can be published, discovered, composed and executed across the Web in a task-driven, semi-automatic way. The step that turns Web services into Semantic Web Services is the semantic enrichment of an existing service description. Representations formalisms for the semantic annotation of Web services include OWL-S [1, 54], WSMO

[8] swordfish.rdfweb.org/discovery/2003/06/codjsform/shell.html

[9] http://www.nue.tu-berlin.de/forschung/projekte/mpeg7/

[30] and SAWSDL [28]. Kerrigan and colleagues [43] propose a methodology for creating Web services' annotations. Apart from this, we are not aware of any other methodologies on this topic. In the following we shortly outline existing tools to semantic annotation of services, which are all manual. The METEOR-S Web Service Annotation Framework (MWSAF) [60] is a framework for the semi-automatic annotation of Web services based on OWL-S. WSMO Studio [21] is a modeling environment designed for WSMO. Besides other features, such as ontology modeling, service discovery, composition, and execution, WSMO Studio supports the semantic annotation of Web services. More precisely, the annotation editor allows humans to manually describe services using the SAWSDL language. WSMT [44] is another Semantic Web Service development environment based on WSMO. WSMT supports the creation of WSMO descriptions, the development of mediation mappings, and the linking with execution environments for executing the services and external systems.

Existing tool support for service annotation requires manual annotation of services; so far, no techniques for automation have been proposed.

3.3 Ontology support

In this section, we present methods and techniques in the areas of ontology learning, ontology alignment, and ontology evaluation. We have chosen those areas because of their relevance for the ontology engineering process.

Ontology learning One can distinguish between ontology learning from unstructured resources, i.e., text, and learning from semi-structured resources, such as folksonomies or UML diagrams. Cimiano [14] presents the ontology learning layer cake, which covers the most relevant types of methods and mentions several tasks in ontology learning. In our survey, we analyzed existing approaches to find commonalities and characteristics.

Maedche and Staab [52] describe a framework for building ontologies supported by ontology learning, which combines machine learning and knowledge acquisition. They propose the following five steps:

1. Import and reuse. In the first step, relevant sources are collected. This can include heterogeneous documents, regardless of their degree of structure. Additionally, a core ontology is used that contains generic and domain-specific terms. This ontology is later extended.
2. Ontology extraction. In the second step, concept learning is performed by gathering new concepts that are extracted from selected sources. To do so one applies mainly natural language processing techniques.
3. Ontology pruning. The aim of this step is to focus the ontology on the target domain by removing irrelevant concepts.
4. Ontology refinement. This step extends the ontology into relations between concepts and other types of ontological primitives.
5. Ontology evaluation. In the final step, the ontology is evaluated in its target application. In case it needs to be improved, the process is repeated starting from the first step.

Aussenac-Gilles and colleagues [3] describe the Terminae method for ontology engineering from texts. The approach combines linguistic and modeling techniques. It is semi-automatic in the sense that it depends on the input of a human “supervisor”. The method consists of three main steps:

1. In the first step, domain resources are gathered. This includes a corpus of text documents, as well as existing ontologies, terminologies or other knowledge structures. The authors recommend the assistance of a domain expert to choose representative texts providing good coverage of the domain and give guidelines for the choice of such corpora.
2. In the second step, a linguistic analysis is performed by applying several natural language processing techniques to the corpus generated in the previous step. This is an iterative process that eventually results into lexical data.
3. This third step is about building a conceptual model based on this lexical data. This involves the definition of concepts and organizing them in a hierarchy. Furthermore, normalization can be carried out using methods such as OntoClean [38] in order to improve the conceptual model.

Simperl and colleagues [66] describe a methodology for ontology learning that is embedded in the ontology engineering process as described by Gomez-Perez and colleagues [35]. They introduce a process model with eight phases and associated activities:

1. Feasibility study. This first phase evaluates whether a learning approach to support ontology development is feasible starting from the requirements specification that was created for building the ontology. Ontology engineers, domain experts, and ontology learning experts collaborate in order to come up with a risk analysis document. The feasibility study can be split into the following sub-tasks:

   a. Specify types of ontologies. This involves defining the characteristics of the target ontology (domain, upper-level, etc).
   b. Identify information sources. Here potentially relevant documents that can be used for learning are collected.
   c. Identify required competencies. Competencies required to carry out the learning process should be identified here in order to compose a suitable team.
   d. Identify stakeholders. This task aims at an analysis of stakeholders.
   e. Identify critical points. This involves describing possible risks and how they are handled.

2. Requirements specification. The actors involved in the previous step specify the requirements for the ontology learning process based on the ontology requirements specification document (ORSD) and the risk analysis document, in order to create an ontology learning requirements specification. This involves the following sub-tasks:

   a. Refine ORSD: In this activity, the ontology requirements specification are adapted to the needs of ontology learning.

   b. Specify information sources requirements: The ontology learning experts defines guidelines and characteristics for the corpus that will be used as input for learning.
   c. Specify tool requirements: Ontology learning experts elaborate on the features of the tools to be used.
   d. Specify personnel requirements. Based on the findings from the previous step, the requirements for the team are identified.

3. Selection of information sources. The corpus of documents is finally selected and configured in this phase. The methodology suggests the following activities to achieve this task:

   a. Search additional information sources. Additionally to what has been identified in previous steps, new, relevant sources are identified.
   b. Evaluate information sources. The collection of information sources are analyzed.
   c. Select and customize information sources. This step fine-tunes and adapts the corpus of documents to the requirements of the learning process, which could include activities such as digitization, formatting, etc.

4. Selection of ontology learning methods and tools. Depending on the learning corpus identified, the actual tools and methods are chosen in this step. This might involve changing the information sources if no suitable tools are available for the corpus at hand. A selection of methods and tools can be achieved with the following three sub-steps:

   a. Search ontology learning tools. This results in a collection of possible approaches.
   b. Evaluate tools usability. Each of the approaches must be analyzed with respect to the information corpus and the requirements specification.
   c. Select learning tools. In this step the final selection of tools is made.

5. Learning preparation. Here ontology learning experts configure the learning tools and prepare the information sources for carrying out the learning process. The result is a tool environment, a documentation, and an execution plan. The following sub-tasks were identified:

   a. Assign tools to sub-domains and information sources. This activity establishes a link from the target ontology fragment, the information sources and the learning tools.
   b. Configure tools. The configuration of the tools is done by the learning expert.
   c. Specify user interaction points. The learning expert also provides guidelines and a plan for which user input is required.
   d. Specify order of tool execution. The order of execution is defined in order to allow for an efficient interaction.

6. Learning execution. In this phase, the actual knowledge is acquired through the operation of the ontology learning tools on the information sources previously collected. This is achieved as follows:

   a. Execute tools.
   b. Provide user input. Semi-automatic approaches might require user input.

   c. Evaluate intermediary results. In order to immediately track cases in which a re-iteration of tools is required, the results of the learning execution are constantly evaluated.
   d. Re-iterate learning execution. If results have to be improved, tools are repeatedly executed.

7. Ontology evaluation. The team analyzes the resulting ontologies against the requirements specification. This step exploits methods for ontology evaluation, as described in a latter section of this article.
8. Ontology integration. The learned ontologies are integrated in the final ontology following methods of ontology integration [19, 62]. Sub-tasks suggested include:
   a. Translate. The ontologies should be made available in the appropriate knowledge representation language.
   b. Integrate ontologies. A methodology or algorithm should be applied to guide the integration process.
   c. Evaluate results. This refers to the evaluation of the integration efforts.

For the analysis of the nature of tasks in ontology learning, we selected the methodology presented by Simperl and colleagues because it is the most detailed one, splitting ontology learning in atomic tasks. We do not consider the early stages that are clearly human-driven in the reminder of the section. These are the feasibility study, the requirements specification, and the selection of ontology learning methods. Integration and evaluation are covered in later sections of this article. Our findings are summarized in Table 4.

Learning preparation The assignment and selection of tools in accordance to the goal of the learning process is a task that fully depends on the human user. Similarly, the configuration of the tools, as well as the specification of user interaction points is human-driven. This also applies to specifying the order of tool execution.

Learning execution The execution of the tools itself is obviously automatic, but some tools require human intervention to resume the execution. The evaluation of intermediary results is assumed to be done manually.

Ontology learning supports the efficient development of ontologies. As already mentioned, there are several points in the ontology development process that allow for computational support. However, the use of ontology learning methods requires per design significant human intervention.

Ontology alignment Ontology alignment is an ontology engineering support activity that allows aligning heterogeneous ontologies enabling interoperability across data

Table 4 Ontology learning.

Task	Nature
Assign tools	Human
Configuration	Human
User interaction	Human
Order of execution	Human
Execution	Computational
Evaluation	Human

Table 5 Classification of approaches to ontology alignment.

Approach	Nature	Required input
HOVY	Semi-automatic	Validation
TransSCM	Semi-automatic	Rules
SKAT / ONION	Semi-automatic	Rules
H-Match	Automatic	
Anchor-Prompt	Semi-automatic	Feedback
OntoBuilder	Semi-automatic	Feedback
MapOnto	Semi-automatic	User-provided alignment, validation
OntoMerge	Semi-automatic	Rules
CtxMatch	Semi-automatic	User input
S-Match	Automatic	
HCONE	All modi possible	User-provided alignment, validation
Moa	Automatic	
ASCO	Automatic	
OMEN	Automatic	User-provided alignment
T-Tree	Automatic	Instances
CAIMAN	Semi-automatic	Instances, validation
FCA-Merge	Semi-automatic	Instances, feedback (lattice)
GLUE	Automatic	Instances, feedback
IF-MAP	Automatic	Instances, user input
QOM	Automatic	Instances
OMAP	Automatic	Instances, training
Xu and Embley's	Automatic	Instances, training
OLA	Automatic	Instances
Falcon-AO	Automatic	Instances
RIMOM	Automatic	Instances

sets and applications. In the following we consider a selection of the most relevant methods and techniques in this area.[10] (as described in [25]). We summarize our findings in Table 5.

Only four of the surveyed approaches are fully automatic. For evidence of the nature of each approach we refer to the authors themselves or to the evaluation by Euzenat and Shvaiko [25]. In the summary column we also make an assessment of the types of inputs of the corresponding algorithms. Wehn user inputs or training are required, the manual effort associated with the algorithm is, of course, higher than when the algorithm relies solely on ontology instances, which are either available or can be generated using machine learning techniques.

Summing up, ontology alignment aims at as much automation as possible. However, best results are achieved with human intervention.

Ontology evaluation The majority of the evaluation approaches proposed in the literature introduce methods to evaluate ontology content, i.e. to estimate the quality

[10] We focus on the tools described in the book by Euzenat and Shvaiko in 2007. We would like to point our readers to the Ontology Alignment Evaluation Initiative http://oaei.ontologymatching.org/2008/results/ that evaluates alignment approaches from a more technical perspective rather than for the required human contribution.

of the conceptual model independently of the fitness of use of the ontology under given circumstances. Gangemi et al. introduce in [32] a theoretical framework for various aspects of ontology evaluation and validation, including functionality and usability. OntoClean [38] is a methodology that relies on four fundamental philosophical notions, namely rigidity, unity, identity, and dependence. ODEval operates on taxonomical structures, allowing the detection of potential inconsistencies and redundancies for RDF(S), DAML+OIL and OWL ontologies [33]. The authors propose a list of general-purpose quality criteria for the content of an ontology. OntoManager focuses on the usage of an existing ontology in a concrete application scenario and pragmatically extrapolates statistic data obtained from monitoring ontology users in order to detect potential errors or limitations of the ontology.[11] Though it is easy to apply by the end-users, the accuracy of the results depends on amount of the monitored data and its accuracy.

Assessing the usability of an ontology in a target application context is addressed briefly in [62]. The authors identify a number of issues relevant on this matter, such as compatible domain or representation language. Another exponent in area of research is OntoMetric [50], a framework for selecting among different ontologies. Provided a set of candidate ontologies, OntoMetric computes a quantitative measure of each of them using a framework of 160 features grouped in several dimensions. After specifying the objectives of the application the ontology engineers build a decision tree containing ontology characteristics required in the application setting. The suitability value of each candidate ontology is computed by comparing its features with the nodes of the decision tree. The usage of collaborative filtering techniques to evaluate reuse candidates has been proposed, for instance, in [12]. The tool WEBCORE [12] uses a series of similarity measures as a baseline to compare a pre-defined Golden Standard ontology with a set of available reusable ontologies, and retrieves the ones most similar to the domain based on user ratings.

OntoClean is supported by several ontology engineering environments, however it requires a deep understandings of philosophical notions. The ODEval method is integrated into the WebODE environment. OntoManager is automatic, but it relies of human labor related to the usage of an ontology in a given setting. The method in [62] is not accompanied by a software tool. Though the authors of OntoMetric propose a tool to guide ontology engineers through the process, this tool requires human input in terms of the definition of the importance of the criteria based on which the assessment is made. WEBCORE relies on a pre-defined ontology and on user ratings. The framework introduced in [32] can be used as a reference for the field of ontology evaluation by ontology engineers. These findings are summarized in Table 6.

In general, one could sum up that when ontology evaluation is defined in very specific terms, machine support is possible and mostly available. For an in-depth assessment, however, human judgement is indispensable. A more careful analysis of ontology evaluation, in particular in the light of the different understandings of the evaluation process, would be necessary in order to identify the best ways to combine human and computational intelligence.

[11] http://ontoware.org/projects/ontomanager

Table 6 Classification of approaches to ontology evaluation.

Approach	Nature
Gangemi's approach	Manual
OntoClean	Mostly manually
ODEeval	Fully automatic
OntoManager	Fully automatic
Pinto's approach	Manual
OntoMetric	Partially automatic
WEBCORE	Partially automatic

4 Discussion

In this section, we summarize the findings from the previous sections, listing those tasks that are inherently human-driven. We then give recommendations for fostering user participation through incentives. We conclude the section with describing a game as an example for an incentivized application for generating semantic content.

4.1 Human contribution in the process of semantic content creation

Based on a survey of the relevant literature of the last two decades of semantic technologies research, we have analyzed and discussed the degree to which human intervention is required to accomplish tasks within the process of semantic content creation.

We summarized the findings of this section in Table 7. Conceptual modeling is a human-driven task. Only very few tasks can be automated and the final modeling decision is always taken by human actors. Automation support is partly possible for collecting relevant terms or for detecting properties of concepts based semi-structured knowledge corpora such as folksonomies.

The semantic annotation of various types of media allows for substantially more automation. The majority of approaches to text annotation leverage natural language processing techniques that have already been subject of investigation for many years now. The best results are achieved when training data is available: usually, a human user has to supervise the machine up to point where the machine is able to

Table 7 Summary of the results.

Task	Nature
Ontology development	Largely human-driven. Collection of terms and definition of class hierarchies can be automatized.
Annotation of text	Semi-automatic with human contribution.
Annotation of multimedia	Largely human-driven with little automation.
Annotation of Web services	Human-driven.
Ontology learning	Learning tools are automatic or semi-automatic (requiring human intervention). Activities around learning are human-driven.
Ontology alignment	Tools are all designed for automation, but the majority required human contribution.
Ontology evaluation	Some aspects can be evaluated automatically. However, whether an ontology is suitable to be used in a domain can to a large extent only be evaluated by a human actor.

reliably perform the task autonomously or provides suggestions. An area that is much younger is the annotation of multimedia content. Many of the existing approaches aim at the extraction of low-level semantics from multimedia. However, the real challenge is the provision of high-level semantics, which refer to the subject of the multimedia artifact. This can only be done to a very limited extent, i.e., by applying machine learning with a vertical focus for a specific domain. In turn, such an approach also requires substantial training. The annotation of Web services is currently a manual task, but more research is needed in order to clearly determine whether this can be traced back to the nature of the task, or to the fact that the corresponding area is not mature enough to produce methods and tools that can feasibly cope with automation.

In the area of ontology learning we have analyzed several methodologies published in the literature of the last years: the execution of ontology learning tools can be done automatically, even though the tools require human input and feedback. The remaining tasks, such as preparation and selection of appropriate tools, are inconceivable without a significant amount of domain expertise. The variety of tools available makes a general assessment difficult. However, as presented in [34], 7 out of 17 tools need user intervention throughout the entire process, 8 can be run semi-automatically, and only 2 are fully automated.

No generic methodologies for ontology alignment have been proposed so far in the ontology engineering community, but there are a multitude of methods and tools that can be used to match, merge, integrate, map and align heterogeneous ontologies. Starting from the list presented by [25], we assessed the need for human intervention in deploying existing ontology alignment technology. Very few tools can be run completely automatically. The remaining ones require the human in the loop to either provide training data or give feedback on the suggestions of the system. Some tools depend on user-defined rules to carry out mappings or expect the user to validate the automatically generated results.

Ontology evaluation is a very broad topic. In its nature it is still human-driven as it has to either evaluate what was initially produced manually in the conceptual modeling phase, or the usability of an existing ontology against a new set of requirements. We have outlined several methods covering both topics. Automatic support is available for checking basic structure-based features of taxonomies, whilst front-end tools are helpful to guide the human evaluator throughout the sometimes complex evaluation procedure. Similarly to the requirements analysis or the conceptualization, ontology evaluation is hardly feasibly automatable for generic domains and application settings, as it requires in depth knowledge in ontology engineering that can not be recorded in a generic tool.

4.2 Motivating user participation

Having identified the tasks that require human intervention, we now discuss possible incentives or incentive-driven tools for motivating user participation. This list is not intended to be complete; it makes preliminary recommendations for potential ways to increase the interest of lay users in semantic content creation. We first discuss monetary and non-monetary incentives [40, 48, 53] as known from the related Web 2.0, and then describe how these incentives can be applied to human-centric semantic content creation tasks as those discussed in the previous sections. We conclude by introducing a casual game through which one can annotate YouTube videos using RDF and ontologies.

4.2.1 Incentives

The advent of Web 2.0 has intensified research efforts studying incentives for Web-based applications [40, 48, 53]. We use the insights of this research as a baseline for our recommendations for incentives for semantic content creation.

Ease of use User-friendly and easy-to-use applications can foster participation while complex tools that take a long time to learn are daunting. This is supported by the success of many Web 2.0 applications that found intuitive ways to represent or gather information and applies to all applications that are built to achieve tasks that are human-centric.

Fun This incentive is mostly related to games. The most popular approach in this context hides a complex task behind casual games [67, 81].

Competition Competition is a relevant incentive in the context of games (rankings), but it can also be a strong driver in community portals where the user with the e.g., most contributions or highest status is awarded.

Reputation Reputation is an important factor within (virtual) communities: it can drive users to invest time and effort to improve the opinion of their peers about them. It has shown that this is an important motivation for, for instance, Wikipedia users [48].

Community sense Staying with the example of Wikipedia as a strong community, studies have shown that the feeling of belonging to a group makes users be more active and supportive. They feel needed and thus obliged to contribute to the joint goals of their community [48].

Reciprocity Reciprocity means that contributors receive an immediate or long-term benefit in return to spending their time and resources on performing a certain task. An example for this is tagging, where the user organizers her knowledge assets, such as bookmarks or pictures.

Monetary incentives Monetary incentives in a Web context can be exploited through a marketplace where human labor is traded. Examples for this are crowd sourcing platforms, such as Amazon's Mechanical Turk[12] or InnoCentive.[13]

4.2.2 Incentives-driven tools

Several options can be imagined for designing incentives-driven tools for ontology development. Games can be deployed for collecting relevant terms, properties, or

[12] https://www.mturk.com

[13] http://www.innocentive.com

for specifying the domain and range of properties. Games can be used for typing a relevant term according to an ontology meta-model, such as OWL or RDFS. For creating a class hierarchy this is already successfully shown in the OntoPronto game [68]. Social incentives can also be equally useful. This has been initiated through projects such as Semantic MediaWiki [79] or myOntology [69]. Another possibility to collect relevant terms by a community of users in order to derive a shared vocabulary is to resort to folksonomies resulting from tagging of resources, thus offering the incentive of immediate benefit of improved knowledge organization through tags. A crowd sourcing platform for building ontologies, or achieving at least specific parts of the process of ontology development, would be another option.

Annotation of text can largely be done automatically given a training set. These training sets can generated using games or can be taken from knowledge intensive portals, such as semantic wikis. For instance, users can be awarded points in a community context for contributing instances to populate ontologies. Similar ideas apply also to the annotation of multimedia or services. However, both tasks require more human input than in the case of textual information. ESP and OntoTube [68, 80] propose casual games effectively harvesting this input. For the annotation of Web services, a marketplace seems more feasible than games due to the very technical nature of the task and the restricted community that can be realistically targeted for annotation purposes.

Ontology learning requires human input during learning preparation or for generating or adjusting training sets. The latter can be achieved using games or through manually annotated corpora of knowledge emerging from community platforms or tagging.

Alignment is suitable for games, especially for generating training sets and for validating automatically generated alignments. Finally, ontology evaluation is a complex enterprize that cannot easily be split up in repeatable tasks to be presented in the form of games. Monetary incentives or a community-driven approach, where a part of a community is responsible for the definition of tests and another part for ensuring the results pass these tests could yield more promising results.

In the following section, we describe the game OntoTube, as an example for "incentivized semantic content creation".[14]

4.2.3 Games for semantic content creation: OntoTube

User-generated video content is constantly increasing on the Web. However, the amount of metadata available for feasibly managing this vast multimedia collection is limited. We assume that rich descriptions of user-generated video content on the Web would significantly improve search over video content.

In the OntoTube game, the players are shown a randomly chosen YouTube video, which starts playing immediately. For each video, the players have to agree on answers for a set of questions derived from a background video ontology. The more questions the players manage to answer consensually, the more points they earn. The number of points depends on the difficulty of the question. Figure 1 show two screenshots of this prototype.

[14] http://www.ontogame.org

Fig. 1 OntoTube game: type of video.

The output of this game is lightweight semantic annotations of videos that can not be gained by applying automatic methods. Players have the incentive of playing a game that is competitive and entertaining and at the same time produce useful semantic annotations. The task of annotation is thereby hidden well behind a motivating concept and an extremely simple interface.

5 Conclusion

Semantic technologies have not reach worldwide adoption yet. One of the reasons for this state of affairs is the limited involvement of users in the process of semantic content creation and maintenance. This can be traced back to a lack of incentives in semantic applications. In the survey we have analyzed which tasks within the authoring and maintenance of semantic content requires human contribution and intervention at various stages. Our findings indicate that especially conceptual modeling is an inherently human-driven task. The same applies to the semantic annotation of multimedia content, which cannot be interpreted by automatic tools. Annotation of Web services is yet not tool-supported, though this might change in the future as the area of Semantic Web Services matures. Ontology support activities also require the human in the loop at many ends: despite some degree of automation, most tools require human intervention, especially when it comes to preparing an ontology learning environment or to evaluating ontologies. We have provided an overview of intrinsic and extrinsic incentives in the area of Web 2.0 that are likely to be applicable to the Semantic Web as well. We have outlined types of applications that could incorporate such incentives in order to address the human-driven tasks identified and described one approach which has already proven to be successful, the OntoTube game for the semantic annotation of videos.

Acknowledgements This work was supported the EU-funded project INSEMTIVES - Incentives for Semantics (www.insemtives.eu, FP7-ICT-2007-3, Contract Number 231181).

References

1. Akkiraju, R., Farrell, J., Miller, J., Nagarajan, M., Schmidt, M.T., Sheth, A., Verma, K.: Web service semantics—wsdl-s. Technical Report, W3C Member Submission (2005)
2. Asirelli, P., Little, S., Martinelli, M., Salvetti, O.: Mulimedia metadata management: a proposal for an infrastructure. In: Proceedings of SWAP 2006 (2006)
3. Aussenac-Gilles, N., Despres, S., Szulman, S.: Ontology learning and population: bridging the gap between text and knowledge. Chapter: The TERMINAE method and platform for ontology engineering from texts, pp. 199–223. IOS, Amsterdam (2008)
4. Barras, C., Geoffrois, E., Wu, Z., Liberman, M.,: Transcriber: development and use of a tool for assisting speech corpora production. Speech Communication Special Issue on Speech Annotation and Corpus Tools **33**(1–2) (2000)
5. Bernaras, A., Laresgoiti, I., Corera J.: Building and reusing ontologies for electrical network applications. In: European Conference on Articifial Intelligence (ECAI'96) (1996)
6. Bloehdorn, S., Petridis, K., Saathoff, C., Simou, N., Tzouvaras, V., Avrithis, Y., Handschuh, S., Kompatsiaris, Y., Staab, S., Strintzis, M.G.: Semantic annotation of images and videos for multimedia analysis. In: European Semantic Web Conference (ESWC). Springer LNCS, New York (2005)
7. Bontcheva, K., Kiryakov, A., Cunningham, H., Popov, B., Dimitrov, M.: Semantic web enabled, open source language technology. In: EACL Workshop on Language Technology and the Semantic Web: NLP and XML, Budapest, Hungary (2003)
8. Brank, J., Grobelnik, M., Dunja M.: A survey of ontology evaluation techniques. In: Conference on Data Mining and Data Warehouses (SiKDD 2005) (2005)
9. Braun, S., Schmidt, A., Walter, A., Nagypal, G., Zacharias, V.: Ontology maturing: a collaborative web 2.0 approach to ontology engineering. In: Workshop on Social and Collaborative Construction of Structured Knowledge at the 16th International World Wide Web Conference (WWW2007), Banff, Canada, 8 May 2007
10. Buerger, T., Ammendola, C.: A user centered annotation methodology for multimedia content. In: Poster Session at ESWC 2008. CEUR, vol. 367 (2008)
11. Campanini, S.E., Castagna, P., Tazzoli, R.: Platypus wiki: a semantic wiki wiki web. In: International Semantic Web Conference (ISWC), pp. 1–6, 10 December 2004
12. Cantador, I., Fernández, M., Castells, P.: Improving ontology recommendation and reuse in WebCORE by collaborative assessments. In: Proceedings of the 1st International Workshop on Social and Collaborative Construction of Structured Knowledge CKC 2007 at the at the 16th International World Wide Web Conference WWW 2007 (2007)
13. Chakarvarthy, A., Ciravegna, F., Lanfranchi, V.: Cross-media document annotation and enrichment. In: 1st Semantic Authoring and Annotation Workshop (SAAW2006) (2006)
14. Cimiano, P.: Ontology Learning and Population from Text: Algorithms, Evaluation and Applications. Springer, New York (2006)
15. Cimiano, P., Handschuh, S., Staab, S.: Towards the self-annotating web. In: Thirteenth International Conference on World Wide Web (2004)
16. Ciravegna, F., Wilks, Y.: Designing Adaptive Information Extraction for the Semantic web in Amilcare, Volume Annotation for the Semantic Web. IOS, Amsterdam (2003)
17. Ciravegna, F., Dingli, A., Petrelli, D., Wilks, Y.: User-system cooperation in document annotation based on information. In: 13th International Conference on Knowledge Engineering and KM (EKAW02) (2002)
18. De Moor, A., De Leenheer, P., Meersmann, R.: Dogma-mess: A meaning evolution support system for interorganizational ontology engineering. In: 14th International Conference on Conceptual Structures (ICCS 2006). Springer LNCS, New York (2006)
19. Dellschaft, K., Staab, S.: Ontology learning and population: briding the gap between text and knowledge. Chapter: Strategies for the Evaluation of Ontology Learning, pp. 253–272. IOS, Amsterdam (2008)
20. Dill, S., Gibson, N., Gruhl, D., Guha, R.V., Jhingran, A., Kanungo, T., Rajagopalan, S., Tomkins, A., Tomlin, J.A., Zien, J.Y.: Semtag and seeker: bootstrapping the semantic web via automated semantic annotation. In: Twelfth International World Wide Web Conference (2003)
21. Dimitrov, M., Simov, A., Momtchev, V., Konstantinov, M.: WSMO studio — a semantic web services modelling environment for WSMO (system description). In: European Semantic Web Conference (ESWC 2007), Innsbruck, Austria (2007)

22. Dingli, A., Ciravegna, F., Wilks, Y.: Automatic semantic annotation using unsupervised information extraction and integration. In: K-CAP 2003 Workshop on Knowledge Markup and Semantic Annotation (2003)
23. Domingue, J., Dzbor, M., Motta, E.: Magpie: Supporting browsing and navigation on the semantic web. In: ACM Conference on Intelligent User Interfaces (IUI) (2004)
24. Ehrig, M., Sure, Y.: Ontology mapping—an integrated approach. Technical Report, AIFB Karlsruhe (2004)
25. Euzenat, J., Shvaiko, P.: Ontology Matching. Springer, New York (2007)
26. Euzenat, J., Mocan, A., Scharffe, F.: Ontology alignments. In: Semantic Web and Beyond, vol. 6. Springer, New York (2007)
27. Falconer, S.M., Storey, M.-A.: A cognitive support framework for ontology mapping. In: Asian Semantic Web Conference (ASWC 2007) (2007)
28. Farrell, J., Lausen, H.: Semantic annotations for wsdl and xml schema. Technical Report, W3C Recommendation (2007)
29. Fellbaum, C.: Wordnet: An Electronic Lexical Database. MIT, Cambridge (1998)
30. Fensel, D., Lausen, H., Polleres, A., De, J., Bruijn, Stollberg, M., Roman, D., Domingue, J.: Enabling Semantic Web Services: The Web Service Modeling Ontology. Springer, New York (2006)
31. Fernandez-Lopez, M., Gomez-Perez, A., Juristo, N.: Methontology: from ontological art towards ontological engineering. In: AAAI97 Spring Symposium Series on Ontological Engineering, Stanford, USA (1997)
32. Gangemi, A., Catenacci, C., Ciaramita, M., Lehmann, J.: Modeling ontology evaluation and validation. In: Proceedings of the European Semantic Web Conference (ESWC 2006), pp. 140–154. Springer, New York (2006)
33. Gómez-Pérez, A.: Evaluation of ontologies. Int. J. Intell. Syst. **16**(3), 391–409 (2001)
34. Gomez-Perez, A.: A survey of ontology learning methods and techniques. Technical Report, OntoWeb Deliverable 1.5 (2003)
35. Gomez-Perez, A., Fernandez-Lopez, M., Corcho, O.: Ontological engineering. In: Advanced Information and Knowledge Processing. Springer, New York (2004)
36. Gruber, T.R.: Toward principles for the design of ontologies used for knowledge sharing. Int. J. Human-Comput. Stud. **43**, 907–928 (1995)
37. Gruninger, M., Fox, M.S.: Methodology for the design and evaluation of ontologies. In: Workshop on Basic Ontological Issues in Knowledge Sharing, Montreal, Canada 1995
38. Guarino, N., Welty, C.A.: Evaluating ontological decisions with ontoclean. Commun. ACM **45**(2), 61–65 (2002)
39. Handschuh, S., Staab, S., Ciravegna, F.: S-cream—semi-automatic creation of metadata. In: SAAKM 2002—Semantic Authoring, Annotation & Knowledge Markup (2002)
40. Hemetsberger, A.: When consumers produce on the internet: the relationship between cognitive-affective, socially-based, and behavioral involvement of prosumers. J. Soc. Psychol. (2003)
41. Holsapple, C.W., Joshi, K.D.: A collaborative approach to ontology design. Commun. ACM **45**(2), 42–47 (2002)
42. Jarrar, M., Meersman, R.: Formal ontology engineering in the dogma approach. In: On the Move to Meaningful Internet Systems 2002: CoopIS, DOA, and ODBASE (2002)
43. Kerrigan, M., Mocan, A., Simperl, E., Fensel, D.: Modeling semantic web services with the web service modeling toolkit. Technical Report, Semantic Technology Institute (STI) (2008)
44. Kerrigan, M., Mocan, A., Tanler, M., Fensel, D.: The web service modeling toolkit — an integrated development environment for semantic web services (system description). In: European Semantic Web Conference (ESWC 2007), Innsbruck, Austria (2007)
45. Kiryakov, A., Ognyanov, D., Manov, D.: Owlim—a pragmatic semantic repository for owl. In: Proceedings of International Workshop on Scalable Semantic Web Knowledge Base Systems (SSWS 2005), WISE 2005, New York City, USA, 20 November 2005
46. Kogut, P., Holmes, W.: Applying information extraction to generate daml annotations from web pages. In: First International Conference on Knowledge Capture 2001
47. Kotis, K., Vouros, G.A.: Human-centered ontology engineering: the hcome methodology. Knowl. Inf. Syst. **10**(1), 109–131 (2005)
48. Kuznetsov, S.: Motivations of contributors to wikipedia. ACM SIGCAS Comput. Soc. **36**(2) (2006)
49. Lenat, D.B., Guha, R.V.: Building Large Knowledge-based Systems: Representation and Inference in the Cyc Project. Addison-Wesley, Reading (1990)

50. Lozano-Tello, A., Gomez-PerezL, A.: Ontometric: a method to choose the appropriate ontology. J. Database Manage. **15**(2) (2004)
51. Lux, M., Becker, J., Krottmaier, H.: Caliph & Emir: semantic annotation and retrieval in personal digital photo libraries. In: CAISE Forum at the 15th Conference on Advanced Information Systems Engineering (2003)
52. Maedche, A., Staab, S.: Ontology learning for the semantic web. IEEE Intell. Syst. **16**(2), 72–79 (2001)
53. Marlow, C., Naaman, M., Boyd, D., Davis, M.: Position paper, tagging, taxonomy, flickr, article, toread. In: In Proceedings of the World Wide Web Conference (WWW2006), Edinburgh, Scotland, 22–26 May 2006. ACM (2006)
54. Martin, D., Burstein, M., Denker, G., Hobbs, J., Kagal, L., Lassila, O., McDermott, D., McIlraith, S., Paolucci, M., Parsia, B., Payne, T., Sabou, M., Sirin, E., Solanki, M., Srinivasan, N., Sycara, K.: Owl-s 1.0 release (2004)
55. Mika, P.: Ontologies are us: a unified model of social networks and semantics. Journal of Web Semantics (Elsevier) **5**(1), 5–15 (2007)
56. Nicola, A.D., Navigli, R., Missikoff, M.: Building an eprocurement ontology with upon methodology. In: Proceedings of the 15th e-Challenges Conferences, Lublijana, Slovenia (2005)
57. Noy, D.L., McGuinness, N.: Ontology development 101: a guide to creating your first ontology. Technical Report, Stanford University, Stanford (2000)
58. Noy, N., Musen, M.: Anchor-prompt: using non-logical context for semantic matching. In: IJCAI Workshop on Ontologies and Information Sharing, pp. 63–70. Seattle (WA US) (2001)
59. Noy, N.F., Musen, M.: The prompt suite: interactive tools for ontology merging and mapping. Int. J. Human-Comput. Stud. **59**(6):983–1024 (2003)
60. Patil, A., Oundhakar, S., Sheth, A., Verma, K.: Meteor-s web service annotation framework. In: World Wide Web Conference (WWW 2004) (2004)
61. Petridis, K., Anastasopoulos, D., Saathoff, C., Timmermann, N., Kompatasiaris, I., Staab, S.: M-ontomat-annotizer: image annotation. Linking ontologies and multimedia low-level features. In: Engineered Applications of Semantic Web Session (SWEA) at the 10th International Conference on Knowledge-Based & Intelligent Information & Engineering Systems (KES 2006) (2006)
62. Pinto, H.S., Martins, J.P.: A methodology for ontology integration. In: International Conference on Knowledge Capture (K-CAP), pp. 131–138. ACM, New York (2001)
63. Popov, B., Kiryakov, A., Kirilov, A., Manov, D., Ognyanoff, D., Goranov, M.: Kim—semantic annotation platform. In: Symposium on Applied Computing, 20–23 October 2005. LNAI, vol. 2870, pp. 834–849. Springer (2005)
64. Reeve, L., Han, H.: Survey of Semantic Annotation Platforms, pp. 1634–1638. ACM, New York (2005)
65. Schaffert, S.: Ikewiki: a semantic wiki for collaborative knowledge management. In: 1st International Workshop on Semantic Technologies in Collaborative Applications STICA 06 (2006)
66. Simperl, E., Tempich, C., Vrandecic, D.: Ontology learning and population: bridging the gap between text and knowledge. Chapter: A Methodology for Ontology Learning, pp. 225–249. IOS, Amsterdam (2008)
67. Siorpaes, K., Hepp, M.: Games with a purpose for the semantic web. IEEE Intell. Syst. **23**(3), 50–60 (2008)
68. Siorpaes, K., Hepp, M.: OntoGame: weaving the semantic web by online games. In: European Semantic Web Conference (ESWC 2008), Teneriffe, Spain. Springer LNCS (2008)
69. Siorpaes, K., Hepp, M., Klotz, A., Hackl, M.: Myontology: tapping the wisdom of crowds for building ontologies. Technical report, STI Innsbruck Technical Report (2008)
70. Souzis, A.: Building a semantic wiki. IEEE Intell. Syst. **20**(5), 87–91 (2005)
71. Specia, L., Motta, E.: Integrating folksonomies with the semantic web. In: ESWC 2007. Springer, New York (2007)
72. Spyns, P., Tang, Y., Meersman, R.: A model theory inspired collaborative ontology engineering methodology. Technical Report, VUB StarLab (2007)
73. Staab, S., Schnurr, H.P., Studer, R., Sure, Y.: Knowledge processes and ontologies. IEEE Intell. Syst. **16**(1), 26–34 (2001)
74. Sure, Y.: Methodology, tools and case studies for ontology based knowledge management. PhD thesis, University of Karlsruhe (2003)
75. Uren, V., Cimiano, P., Iria, J., Handschuh, S., Vargas-Vera, M., Motta, E., Ciravegna, F.: Semantic annotation for knowledge management: requirements and a survey of the state of

the art. Web Semantics: Science, Services and Agents on the World Wide Web **4**(1), 14–28 (2006)
76. Uschold, M., King, M.: Towards a methodology for building ontologies. In: Workshop on Basic Ontological Issues in Knowledge Sharing, Montreal, Canada (1995)
77. Van Damme, C., Hepp, M., Siorpaes, K.: Folksontology: An integrated approach for turning folksonomies into ontologies. In: Proceedings of the Workshop Bridging the Gap between Semantic Web and Web 2.0 at the ESWC 2007, Innsbruck, Austria, 7 June 2007, p. 15. LNCS Springer (2007)
78. Vargas-Vera, M., Motta, E., Domingue, J., Lanzoni, M., Stutt, A., Ciravegna, F.: Mnm: ontology driven semi-automatic and automatic support for semantic markup. In: 13th International Conference on Knowledge Engineering and Management (EKAW2002) (2002)
79. Völkel, M., Krötzsch, M., Vrandecic, D., Haller, H.: Semantic Wikipedia. 23–26 May 2006
80. Von Ahn, L., Dabbish, L.: Labeling images with a computer game. In: CHI 2004. ACM, New York (2004)
81. von Ahn, L., Dabbish, L.,: Designing games with a purpose. Commun. ACM **51**(8), 58–67 (2008)
82. Vrandecic, D., Pinto, S., Tempich, C., Sure, Y.: The diligent knowledge process. J. Knowl. Manag. **9**(5), 85–96 (2005)