Semantics through Language Sharing

Jessica Rubart
arvato services1
Gütersloh, Germany
info@jessicarubart.de

ABSTRACT
This paper discusses how existing approaches in the hypermedia and software engineering fields support web semantics. Web semantics is an important issue in the context of Web Science. Both directions, the Semantic Web and Folksonomies, focus on language sharing. It is argued that to build holistic hypermedia systems these language kinds need to be integrated or mapped. A web semantics space is proposed that supports comparison of approaches with respect to language sharing.

Categories and Subject Descriptors

General Terms
Design, Human Factors, Languages

Keywords
Web Science, web engineering, web semantics, language sharing

1. INTRODUCTION
Web Science is a research agenda focusing on the analysis and synthesis of web-like technologies, its use and governance [2]. In this regard engineering issues are one important part of Web Science focusing on the web’s development and its requirements. Web semantics plays a significant role in the context of web engineering. It addresses standards, languages, methods, models, etc. to support the understanding between decentralized information systems and users. On the one hand side there is the Semantic Web [3] focusing on smart integration of machines. With the Semantic Web users can not only find material, but also infer additional information not directly related. And on the other hand side there is the Web 2.0 focusing on the web as a platform and in particular on participation and tagging [28]. People collaborate while creating content. Folksonomies emerge as user-created languages for categorization. Given these two extremes, what is it that constitutes semantics and that these two directions have in common?

This position paper aims at discussing how existing approaches in the hypermedia and software engineering fields support understanding, whether the focus is on human sense making or on powerful and smart software, and whether solutions are provided to integrate these different language directions to form holistic hypermedia systems.

2. HYPERMEDIA FIELD

2.1 Semantic Web
The Semantic Web focuses on the integration of machines. They shall be able to interpret and reason about data across applications and organizational boundaries. Based on this focus, languages, such as the Resource Description Framework (RDF) [13] and the Web Ontology Language (OWL) [15], have been defined. Those languages and concrete instances are shared between machines, more precisely software components running on machines. They are not intended for end users. Therefore, strong tools are required to provide a mapping from the end users’ language(s) to the machine representations.

2.2 Folksonomies
The term “folksonomy” was coined by Vander Wal in 2004 as a combination of “folk” and “taxonomy” [30]. It refers to free tagging of information resources by end users and sharing tags with others. Usually, tags can be used for queries. These tags describe emerging semantics as there is no predefined ontology. Compared to ontologies collaborative tagging is informal, quick, and with low barriers [2]. A user community can cultivate their tagging language. This integrates users, but complicates machine-side usage of the classification. There are solutions to some particular problems in this area, such as WordFlickr [12]. It uses a semantic lexical database for improving queries with respect to images in Flickr.

2.3 Spatial Hypertext
Spatial hypertext [14] supports the creation of structure, such as categories and relationships, by means of visual attributes and spatial proximity. This helps the end user to express emerging structure. It is particularly useful for supporting an incremental knowledge building process [27]. The visual knowledge builder (VKB) provides suggestions on how to formalize and organize the structure. In this way spatial hypertext focuses on emerging
structure and thus on an emerging structuring language from the end user’s point of view. The pro-active suggestion agent approach, however, focuses on individual usage; it does not address group decisions. Identified types that are known to the system can be processed by the machine.

2.4 Schema-based Hypertext
Schema-based hypertext utilizes typed hypermedia elements and can place constraints on allowed instances. These types can either relate to system types, such as in MACWEB [17] or RDF [13], or to semantic types, such as in Compendium [5]. Semantic types describe the semantic purpose of a structure element and are also referred to as “role” [18] or “semantic role” [6]. Semantic types can represent the users’ language whereas system types can represent the systems’ language. In [25] an approach is described that allows mapping of system types to semantic types by collaborative configuration. Approaches like this map the different language kinds to provide for holistic application support.

2.5 Structural Computing
Structural computing [19] is concerned with many different structural abstractions. The Web 1.0 is an example of navigational hypermedia aiming at supporting the creation and navigation of information spaces. In addition to the navigational domain structures like argumentation, spatial, metadata, and taxonomic ones have been identified and supported by structural computing environments. While structural computing environments focus on infrastructure and middleware the different structural abstractions, i.e. hypermedia domain [29], focus on solving particular organization problems from the users’ point of view. Therefore, these can be seen as users’ languages that are supported by structural computing environments providing for the foundations to support the structural abstractions. Semantics can be described by structural computing applications, which use the middleware services in a structural computing environment so that end users can create useful structures. Dependent on the hypermedia domain, those structures can describe semantics either explicitly, e.g. taxonomic structures, or implicitly.

2.6 Workflow Domain
Workflow technology is intended to support work by enacting explicitly modeled and represented business processes. Cooperative hypermedia solutions have been developed that integrate process models [24], [31]. To model processes, task-related semantics, such as information about a task’s state, have been modeled into a special composite class. Analogous, control and data flow semantics have been integrated into a special link class. With this, one could express, for example, which nodes of a source task shall flow into a destination task composite after completing the source task. Workflow support can be integrated as a workflow service in the multiple open services approach [32] and, thus, as a structure server on the middleware level of a structural computing environment. In this way it can support structural computing applications in expressing workflow semantics.

In the workflow field there are different approaches with respect to interoperability. The Workflow Management Coalition (WfMC) [33] has defined a terminology, a workflow reference model, and interfaces to a workflow engine, which is responsible for workflow execution. This work mainly concentrates on interoperability between systems. Similarly, the Web Services Business Process Execution Language (WSBPEL) [20] focuses on a language for business process orchestration based on open web services. The Business Process Modeling Notation (BPMN) [22] is a standardized graphical modeling language for users expressing workflows. To provide holistic support for workflow execution, one needs to bear in mind that a model activation process can be [4]:

- **Automated**, where a software component interprets the model,
- **Manual**, where the model guides the actions of human actors, or
- **Interactive**, where prescribed aspects of the model are automatically interpreted and ambiguous parts are left to the users to resolve.

By updating such a process model interactively, users can adapt the system to fit their local plans, preferences and terminology [11]. Interactive model activation includes the interaction between the participants and the workflow engine as well as the interaction among the participants themselves [26].

3. SOFTWARE ENGINEERING FIELD

3.1 Patterns
Pattern languages define a shared vocabulary. Each pattern names and describes a recurring problem and a reusable solution. Initially, patterns have been applied to buildings and towns [1]. Regarding reusable object-oriented software, design patterns have been presented [8]. A design pattern abstracts key aspects of a common design structure. It is based on a practical solution that has been applied several times in different systems. Therefore, a design pattern represents a proven solution for a recurring problem. It supports basing new designs on prior experiences. Thus, hypermedia design patterns can support a shared vocabulary and shared understanding between developers focusing on sharing the system-oriented language between humans. When using domain-driven design (see below) the terminology of the domain is used by developers as well as customers and should be used in domain-oriented design patterns. There are different categories for patterns in the software domain, such as design patterns, architectural patterns, or analysis patterns. Usage-oriented patterns, such as one for describing podcasting, support sharing a user-oriented language.

3.2 Model-Driven Architecture
The Model-Driven Architecture (MDA) addresses interoperability between models [21]. A model represents a part of a system. The MDA separates business and application logic from the technical platform. So-called Platform-Independent Models (PIMs) focus on the business and application logic, whereas Platform-Specific Models (PSMs) tackle the technical modeling issues. PSMs are created using model-to-model transformations. Code is generated based on PSMs. Therefore, MDA points out a domain-specific model that takes into account the customers’ language and gets mapped into the technical part of the system.
3.3 Domain-Driven Design

Domain-Driven Design (DDD) focuses on the understanding of the business domain [7]. The business domain is the core complexity that should be captured in the domain model. In this way DDD leads to model-driven design. The domain model provides a shared language known by the customers and the development team. It provides a bridge between the user-oriented language and the system-oriented one. It is actually part of both and called a ubiquitous language. From the perspective of a conceptual architecture the domain layer is between the infrastructure layer and the application layer focusing on core domain issues.

4. CONCLUSIONS

Web semantics is an important issue in the context of Web Science. Both directions, the Semantic Web and Folksonomies, focus on language sharing. While the Semantic Web defines languages for integrating machines, such as RDF [13] and OWL [15], Folksonomies focus on user-created languages. It is these languages that enable understanding between systems or users. To build holistic hypermedia systems these languages need to be integrated or mapped as shown by Figure 1. This enables, for example, full queries on a social classification.

Figure 1. Semantics through Language Sharing

There are approaches to integrate or map these language kinds for particular problems and languages, such as in WordFlickr [12] for improving queries with respect to images in Flickr. General approaches can be divided into ones that map languages and ones that integrate languages, such as domain-oriented ones.

From the viewpoint of flexible information systems interaction can be a useful design metaphor [11]. Designers should look at ways in which users and systems can jointly solve problems rather than try to automate everything. This requires an approach where system languages need to be more human. Thus, those language integration aspects are very important for flexible information systems.

Figure 2 shows sharing the different language kinds (between systems and users respectively) and language mapping as different dimensions of a web semantics space. The higher the support for one direction the more semantics can be expressed regarding that direction. Approaches can be sorted in to compare them with respect to these directions. This space is complementary to the design space for information structuring systems in [9] and the structural diversity space in [10] as it focuses on sharing languages between multiple systems or users and language mapping, which are very important dimensions for web semantics.

The Semantic Web (SW), with its languages, such as RDF and OWL, supports assertions about resources accessible on the Web, ontologies, inference rules, etc. so that systems can interpret and reason about data across applications. This supports very much sharing a language between systems.

An application based on the concepts of the Semantic Web, such as OntoPortal (OP) [16], can support the other directions as well. OntoPortal uses ontological hypertext to provide intelligent navigation of research metadata. On the user interface query by linking makes use of the ontological knowledge and therefore provides a language for the end users and maps it to the system.

Folksonomies describe emerging semantics with respect to a user community who can cultivate their tagging language. An application in this area, such as WordFlickr [12], can support other directions as well. In WordFlickr the language of the users is mapped to the system language by integrating a semantic lexical database.

Figure 2. Web Semantics Space with Semantic Web, Folksonomies, and Applications

Figure 3 shows structural computing concepts in the proposed web semantics space. Services of the foundation or middleware layer in a structural computing environment do not focus on any kind of language sharing or mapping. With open protocols, such as the open hypermedia protocol for the navigational domain [23], structural computing services can support interoperability and therefore sharing languages between systems. With structural computing applications using the middleware services users can create and share structures and hence establish a structuring language. This can potentially happen across applications dependent on the structural computing environment and its support for open protocols.

Figure 3. Structural Computing in the Web Semantics Space
5. REFERENCES


