Towards the Adaptive Web using Metadata Evolution

Nicolas Guelfi
Laboratory for Advanced Software Systems, University of Luxembourg
6, rue Coudenhove-Kalergi L-1359 Luxembourg-Kirchberg (Luxembourg)
Phone number: (+352) 46 66 44 5251
Fax number: (+352) 46 66 44 5500
e-mail: Nicolas.guelfi@uni.lu

Cédric Pruski*
Laboratory for Advanced Software Systems, University of Luxembourg
6, rue Coudenhove-Kalergi L-1359 Luxembourg-Kirchberg (Luxembourg)
and
LRI-PCRI, CNRS-INRIA (Futurs), University of Paris-Sud XI
Parc club Orsay Université, 4 rue Jacques Monod, 91894 Orsay (France)
Phone number: (+352) 46 66 44 5455
Fax number: (+352) 46 66 44 5500
e-mail: cedric.pruski@uni.lu, cedric.pruski@lri.fr

Chantal Reynaud
LRI-PCRI, CNRS-INRIA (Futurs), University of Paris-Sud XI
Parc club Orsay Université, 4 rue Jacques Monod, 91894 Orsay (France)
Phone number: (+33) 1 72 92 59 87
Fax number: (+33) 1 60 19 69 63
e-mail: chantal.reynaud@lri.fr
ABSTRACT

The evolution of Web information is of utmost importance in the design of good Web Information Systems applications. New emerging paradigms, like the Semantic Web, use ontologies for describing metadata and are defined, in part to aid in Web evolution. In this chapter, we survey techniques for ontology evolution. After identifying the different kinds of evolution the Web is confronted with, we detail the various existing languages and techniques devoted to Web data evolution, with particular attention to Semantic Web concepts, and how these languages and techniques can be adapted to evolving data in order to improve the quality of Web Information Systems applications.
Towards the Adaptive Web using Metadata Evolution

When we consider the evolution of the World Wide Web (WWW) and its development over the last decade, we can see drastic changes have taken place. Originally, the Web was built on static HTML documents and used as experimental communication means for specific communities in educational institutions and government defence agencies. With its ever increasing popularity, the WWW is now the largest resource of information in the world. As such, it contains either durable or volatile information that needs to be maintained easily and often with predefined time delays. For example, this is the case for day-to-day news diffusion. The emergence of new paradigms like the Semantic Web (Berners-Lee, Hendler, & Lassila, 2001) has further intensified the evolution of the Web. Since Semantic Web has the ability to make Web information machines understandable, it correlates to an improvement in quality of Web content and Web services and the result unloads users of tedious search tasks.

These changes are accompanied by the manifestation of new application families such as Web Information System (WIS) applications. WIS applications are developed based on Web content and consequently must adapt to its evolution. Consider, for example, a tourism application for helping users prepare their holidays. If the company introduces a new kind of vacation (e.g., social holidays for elderly people) then the application needs to be maintained in order to cope with the evolution of this domain. The development of WIS applications is not an easy task since it is difficult to clearly catch the semantics of actual Web content, which in turn limits the retrieval of relevant Web information (Guelfi & Pruski, 2006).

Parallel to this development, the use of metadata to drive the advancement of Web-based applications has proven to be a very promising field of research (“Web Engineering: Principles and Techniques,” 2005) and we believe that the use of metadata can improve the quality of WIS applications. Metadata is useful to describe Web resources in a structured and precise way. It can be used, for instance, to describe the content of tourism resources in our application example. Consequently, languages are needed to define metadata and techniques are required for the metadata to smoothly follow the evolution of the described resource. To this end, the Semantic Web paradigm can be the key to success. It consists in using ontologies (Gruber, 1993) to define the semantics of a particular domain and then uses the vocabulary contained in these ontologies as metadata to annotate Web content.

In this context, many ontology languages have been intensively studied in the literature (Antoniou & van Harmelen, 2004; Charlet, Laublet, & Reynaud, 2004). Although they provide basic modelling features, they do not offer enough properties to design and maintain ontologies that have the ability to evolve over time. Static ontologies designed with existing languages seem to be unsuitable for today’s dynamic Web and what it requires instead are “dynamic” ones, which are built using new kinds of languages and techniques in order to give them the property of being evolvable over time according to modification of Web content. Since these new languages and techniques will have an important impact on Web-based applications and mainly WIS applications, we need to carefully identify the different kinds of evolution that can affect Web content and to evaluate the current capabilities of existing languages.

In this chapter we propose to survey state-of-the-art dynamic ontology languages and techniques. This study also contains notes concerning knowledge evolution in other paradigms.
connected to the WWW, like reactivity (Alferes et al., 2004) and Adaptive Hypermedia Systems (AHS) (Brusilovsky, 2001), both of which have elements that should be integrated in the development of advanced WIS applications. Such adaptation techniques can be used, for example, to adapt the presentation of content to users. Imagine our tourism application able to display in bold red font a particular offer, like a fee reduction on a particular service. In order to cope with various kinds of Web content evolution, we will provide some elements regarding the improvement of these languages. Based on our survey and the proposals we make, we believe that it will be easier to understand the impact of languages and technologies on the quality of future WIS applications.

We first discuss existing types of evolution impacting the Web as well as related work in the field of languages and techniques for Web data evolution. Dealing with the evolutionary aspect of Web content, two important research fields stand out: reactivity and AHS. Within the framework of reactivity, several interesting Event-Conditions-Actions (ECA) languages have been proposed to not only make Semantic Web concepts reactive but also classic web sites. AHS is a young research field that we will talk about in detail since it takes into account knowledge evolution and introduces user adaptation, which is important for future versions of the Web.

Following this, we discuss related work in the field of languages and techniques for ontology evolution of the Semantic Web. Although a set of standards have being established by the World Wide Web Consortium (W3C) and intensively studied in literature, new ones are under investigation. In this section we will present these emergent languages and techniques. We will briefly introduce standards like Web Ontology Language (OWL) and focus on the description of new languages like dOWL and the presentation of methodologies that have been proposed to deal with ontology evolution. We also propose some perspectives about the extension of the studied languages with respect to ontology evolution.

The next section discusses the impact of the concepts we have presented on the quality of WIS applications. Based on our remarks, we illustrate what has to be done to increase the quality of WIS applications at each stage of its lifecycle, i.e. from design to runtime, mainly by taking into account user preferences, experiences and behaviour. Finally, we wrap up with concluding remarks.
WEB DATA EVOLUTION: PRINCIPLES AND TECHNIQUES

Web Data Evolution

Before addressing the technical aspects of Web data evolution it is important to understand how evolution occurs on the Web and categorize the different kind of evolution (see Figure 1). Evolution depends mainly on two things: the specificities of data and the domain data is related to. In order to understand how evolution depends on these two characteristics, several aspects need to be considered.

First, new information can be added or removed from the Web. This is a basic situation that occurs. Web users or developers build a Web site (or Web application) and then publish it on a Web server. The site is then indexed in a Web search engine by a crawler and can be retrieved when searching the Web. The addition to or removal of information from the Web impacts the evolution of content but also Web morphology, an aspect that is often integrated into Web search applications (Page & Brin, 1998).

Second, once published, content often has to be modified according to events that happen in real life. In this instance, we are referring to data update. Depending on the Web site’s domain of concern, updating data can be done manually if the changes occur periodically, or semi-automatically if the changes are incessant. We observe that the rates of change are strongly correlated with the domain of the data. Domains can be grouped into two different classes according to their frequency of changes: a domain concerning entertainments where changes are infrequent we call a “hobby Web;” and a business domain, “professional Web,” with content undergoing incessant evolution. These classes also differ in the specificities of their data. The content of the hobby Web concentrates various formats of data that is directly accessible. In addition to rough text, multimedia files like videos, sound or images can generally be found. On the contrary, the professional Web contains more room dedicated to textual information with its content often hidden. This is due to the fact that business companies always have an Information System (IS). Historically developed separate from Web technology (client/server application or mainframes, basic computer networking, etc.), the IS, through its Web publication layer, offers information and services to users connected on the Web. This is the current status for WIS applications, even if some modern companies develop the IS as a WIS application. In this case, information is hidden from non-administrative Web users. Keeping information hidden is also necessary for nonfunctional requirements (security, overloading control, etc.), as well as for functional requirements (human resource management, detailed financial data, etc.). As a result, the published area of the professional Web is dynamically generated via the use of scripts that are embedded into the pages source code. In order to cope with the evolution of the data of these WIS applications, developers can use different technologies tailored to the type of content they want to publish. Among the most promising approaches are:

1. XML-based technologies that allow web documents to be structured and its extension with metadata allows data exploitation to be optimized (see next section).
2. Event-Condition-Action (ECA) languages that describe the reactivity of a Web resource when special events (that reflect IS data evolution) occur.

The third aspect to consider in Web evolution is the presentation of the content. It is orthogonal to the two previous aspects and represents the adaptability of the data presentation to a Web user when data evolves. This implies that the user interface reflects the data evolution. In
In this case, current implementation uses a metadata-like approach to better present data to the user (see below AHS).

Therefore, to tackle the various kinds of evolution that are presented in this section, new techniques, proposed in the context of young research fields like reactivity and adaptive hypermedia, have been developed.

**Reactive Web Sites**

Reactivity on the Web is a new emerging paradigm that covers: updating data on the Web, exchanging data about particular events (such as executed updates) between Web sites and reacting to combinations of such events. Reactivity plays an important role for upcoming Web systems such as online marketplaces, adaptive Semantic Web systems, as well as Web Services and Web Grids. Research on reactivity is particularly studied in the recently created REWERSE network of excellence. In this context Web evolution is described as an update or more generally a modification of several individual Web resources. The modifications are the result of events that occur in time and may be triggered by user behaviour. Events are described using ECA languages (Patrânjan, 2005; Schaffert, 2004) and this introduces a new kind of Web content evolution: Conditional Evolution. Consider an e-commerce application that allows the purchase of products. As soon as a user buys the last product available (event), the product must be removed from the set of products offered on the Web site. ECA languages allow the definition of rules for this, which are dynamically interpreted at run-time.

**Adaptive Hypermedia Systems**

Adaptive hypermedia (Brusilovsky, 2001) is a young research field on the crossroads of hypermedia and user modelling. This emerging technology addresses the evolution of the presentation of the data and not directly the evolution of the data itself. Adaptive hypermedia systems build a model of goals, preferences and knowledge of each individual user, throughout the interaction with the user, in order to adapt to the needs of that particular user. Studies in adaptive hypermedia rely on the following observations: traditional hypermedia offers the same Web pages (or more generally speaking, information) to all users whatever their background may be and whatever kind of data evolution has taken place. AHS integrates both user data and behaviour and data evolution to provide the user with an adapted and personalized presentation of data.
We now need to do an in-depth study of the specificities of metadata and in particular the techniques, methods and languages that support its definition to understand its impact on the development of good WIS applications. Metadata is data that describes other data. The term is common in various fields of computer science; however, metadata is of special interest in information retrieval and Semantic Web applications. Although several definitions were originally proposed, each of them concerned a particular domain of computer science. Generally, a set of metadata describes a single set of data, i.e. a resource. There are two distinct classes of metadata: structural (or control) metadata and guide metadata. Structural metadata is used to describe the structure of data such as tables, columns and indexes in databases. Guide metadata is used to help users with information retrieval tasks; for instance a date might be useful to eliminate out-of-date data. Metadata may be expressed as a schema of data of a resource or, in its simplest form, as a set of keywords in a structured language whose aim is to better interpret data. Consider the data “L-1359.” In its raw form L-1359 does not have a precise meaning, but if the metadata “zip code” is added (Luxembourg zip code = country’s first ISO letter, hyphen, and 4-digit town), the data clearly denotes the city Luxembourg. Because of metadata a better interpretation of data is used to:

1. Semantically enhance the capability of retrieving resources. Metadata may be directly visualized by the end-user or may be exploited by automated services and the user unaware of its use. In both cases, metadata allows targeted and more relevant information retrievals.
2. Reduce the complexity of data exploitation through the use of metadata as filters to optimize searching tasks, or via the definition of new compression algorithms for storing purposes, or to automate workflows.
3. Adapt or personalize data presentation. Metadata is intended to enable variable content presentations. For example, if a picture viewer knows the most important region of an image (e.g., where there is a person in it) it can reduce the image to that region and show the user the most interesting detail on a small screen, such as on a mobile phone. A similar kind of metadata is intended to enable blind people “reading” diagrams and pictures (e.g., by converting them for special output devices or by reading a description using voice synthesis).

A detailed illustration of the usefulness of metadata can be found in the Semantic Web initiative (Berners-Lee, Hendler, & Lassila, 2001). The intent of the Semantic Web is to enhance the usability and usefulness of the Web and its interconnected resources through:

1. Documents “marked up” with semantic information (an extension of the HTML <meta> tags used in today’s Web pages for supplying information to Web search engines using web crawlers). Metadata can focus on a specific part of a document consulted by a user (e.g., a metadata defined to indicate a web site administrator’s email and how many times the page has been visited). Metadata can also be indicators about the location and access mode to a document (e.g., the hyperlink nature of a text that allows access to another web page or the need for a specific tool to consult displayed information described textually).
2. Metadata with terms extracted from ontologies (Gruber, 1993) that define it and establish mappings with terms from other ontologies. This allows the ability to exploit Web data even if it has been created in a different context (e.g., one user looking for a book on painting may consider the writer as an “author,” while another user may consider the writer as an “artist”).
3. Software to help end-users of the Web performs tasks using metadata.
4. Web-based services providing information to software agents or web services (e.g., an indication of the quality of service: availability, efficiency, security).

Metadata can be distinguished by:

1. Content. Metadata can describe the resource itself with different focuses like, name and file size, or the content of the resource (e.g., “Video of artist painting.”) The focus can range from the raw level to an abstract one depending on the functional (or nonfunctional) requirements the metadata must fulfill.

2. Evolution. The metadata definition as well as its value may change during its life cycle (e.g., a file is characterized by its name and location, but later we may need to see the file with its extension code (doc, xls, etc.). Or, we can name and rename a file to reflect the transformation of its updates (Paper_v1, Paper_v2, Paper_v3)).

The Semantic Web aims at giving definition to the Web. However, Web data is constantly evolving and therefore, if metadata from a given ontology describes this changing content, the metadata itself must evolve otherwise the semantic of the described data can be erroneous. In most cases, Web data evolution is automatic. Ideally, the evolution of metadata should occur in the same manner. Moreover, the evolution of the ontology (i.e. the domain) will impact the evolution of metadata. Since, concepts can be added to or remove from a given ontology forming then a new vocabulary, metadata that use this “new” ontology has to adapt to these changes. We present, in next section, the various languages and techniques that have been proposed to support ontology evolution.

Techniques and Methodologies for Managing Changes in Ontologies

Changes in ontologies is a topic that has not been intensively studied. However, the existing languages and techniques for ontology engineering introduce two different concepts: ontology evolution and ontology versioning. The former, according to Ljiljana Stojanovic, is “the timely adaptation of an ontology to the arisen changes and the consistent propagation of these changes to dependent artefacts” (Stojanovic, 2004, p. 15). The latter approach is a stronger variant of handling changes to ontologies: ontology versioning allows access to data through different variants of the ontology. In addition to managing the individual variants of the ontology themselves, it is important to also manage the derivation relations between the variants. These derivation relations then allow definition of the notions of compatibility and mapping relations between versions, as well as transformations of data corresponding to the various versions. Versioning is an approach that can be found mainly in languages for managing changes in ontology. This will be presented in the next section.

In Michel Klein’s approach for ontology evolution (Klein, 2004), he proposes a framework (Klein & Noy, 2003) to solve problems relating to ontology evolution. This framework has been developed based on a study of the context of the ontology evolution problem and a comparison with solutions provided by related areas (Klein, 2001; Noy & Klein, 2004). The framework also provides a methodology (Stuckenschmidt & Klein, 2003) described as a change process which is tool supported (Klein, Fensel, Kiryakov, & Ognyanov, 2002). The proposed process describes the different steps in ontology evolution as well as the different problems that arise from the latter. Klein also focuses on the different kinds of evolution that can interfere in ontology evolution. He first points out changes in the conceptualization (the domain can change) then, changes in the specification (the way for describing a domain can differ) and lastly, changes in the languages used to model the domain. An important consideration is how to
capture these changes. The proposed framework is based on the ontology of change operations for providing a formal description of the ontology modifications to be applied. However, as there is no limit on the number of composite operations that can be considered, there is no guarantee that the composite operations cover all needs.

On another front, Ljiljana Stojanovic’s work focuses on a methodology for ontology evolution (Stojanovic, 2004). The proposed methodology (Stojanovic, Maedche, Motik, & Stojanovic, 2002) can be divided into six different steps occurring in a cyclic loop:

1. Change capturing: This consists in the discovery of changes. This task could be done manually by a knowledge engineer or automatically by using any existing change discovery method. Three types of change discovery are defined (Stojanovic, 2004): structure-driven, usage-driven and data-driven. Whereas structure-driven changes can be deduced from the ontology structure itself, usage-driven changes result from the usage patterns created over a period of time. Data-driven changes are generated by modifications to the underlying dataset, such as text documents or a database representing the knowledge modelled by an ontology.

2. Change representation: Before changes are treated, they have to be represented in a suitable format according to the ontology model that is used.

3. Semantics of change: Possible problems that might be caused in the ontology by the changes are determined and resolved. For example, we have to decide what to do in the instance of a removed concept. This stage is useful for checking the consistency of the ontology.

4. Change implementation: In this step, the knowledge engineer is informed about the consequences of the changes, changes are then applied and tracking of performed changes are kept.

5. Change propagation: To propagate the changes to other related ontologies or software agents. The task of the change propagation phase is to ensure consistency of dependent artefacts after an ontology update has been performed. These artefacts may include dependent ontologies, instances, as well as application programs running against the ontology.

6. Change validation: The validation of the evolution process according to the semantics of changes defined (Step 3). This step can initiate additional new changes that need to be performed. In this case, we start over by applying the change capturing phase of a new evolution process.

A methodology, H-CHANGE, is also proposed in the context of P2P systems (Castano, Ferrara, & Montanelli, 2006). This methodology has been conceived specifically for the local evolution of peer ontology and for evolving independent ontologies in open contexts, where distributed concept definitions emerge dynamically through interactions of independent peers. The methodology integrates semi-automated change detection techniques based on semantic matchmaking and change assimilation techniques for evolving ontology. Two strategies are applied as techniques for assimilation: assimilation-by-merging or assimilation-by-alignment according to the level of semantic affinity of the new incoming concept. The choice of the strategy is automated. It is performed according to a threshold-based mechanism.

The complete ontology evolution process is generally not supported by existing tools, but some tools (Haase & Sure, 2004) provide specialized features like, change discovery, keeping track of ontology changes, support for evolution strategies, undo/redo operations, etc. These tools aim at helping users perform the changes manually rather than automatically.

Finally, the problem of ontology evolution is considered a special case of a more general problem, belief change. Some of the most important concepts of belief change have been revised
(Flouris, Plexousakis, & Antoniou 2006) to apply them to the ontology context evolution. On another side, ontology evolution has also been addressed in different languages. These ones are detailed next section.

**Languages for Dynamic Ontology**

**ECA (Event-Condition-Action) Languages for the Semantic Web**

Reactivity for the Web is an important component of the vision of Semantic Web as a closely intertwined network of autonomous nodes. In contrast to the current Web where many sites only provide information and others simply query them on demand, the Semantic Web will profit from enhanced communication between its nodes, not only for answering queries, but also in its evolution. It is crucial that relevant changes to information used by a Semantic Web agent are consistently and rapidly propagated to all interested parties. In this context, ECA languages like Resource Description Framework Triggering Language (RDFTL) (Papamarkos, Poulouvasilis, & Wood, 2004) have been developed. RDFTL is tailored to Resource Description Framework (RDF) data (resources, properties and statements organized in graph). It works on RDF graph and allows nodes or arcs of this graph to be removed, added or updated on defined events occurring on particular RDF statements.

**Simple HTML Ontology Extension (SHOE)**

The first initiative concerning the definition of an ontology language tailored for the Web has been Simple HTML Ontology Extensions (SHOE) language (Heflin & Hendler, 2000). SHOE has an XML-based syntax and semantics (Heflin, Hendler, & Luke, 1999) and is a frame-based language built on top of XML that can be easily integrated into HTML documents. It was designed with the aim of integrating machine-readable semantic knowledge in Web documents. One of the main innovations proposed by this language was to use URI to identify concepts, which have been integrated in the definition of DAML+OIL. Another interesting point is that SHOE has the ability to adapt to the evolution of Web data and thus, several primitives have been defined, such as ontology importation (USE-ONTOLGY tag), local renaming of imported constants (DEF-RENAME tag) and ontology versioning (VERSION tag). Despite these interesting points SHOE suffers from a lack of formalization of the primitives introduced.

**Evolution Aspects in W3C Standards**

With RDF and RDFS the W3C has adopted a set of standards devoted to ontology engineering, but it is only since the release of Web Ontology Language (OWL) (McGuinness & van Harmelen, 2004) that changes in ontologies have been tackled. OWL provides a set of six tags for managing ontology versioning. Two of them directly concern the version of the ontology. In fact, owl:versionInfo and owl:priorVersion give information about the current and the previous version of the changing ontology. Two other tags, owl:backwardCompatibleWith and owl:incompatibleWith, contain statements concerning other ontologies that are compatible or incompatible with the current ontology. Lastly, owl:DeprecatedClass and owl:DeprecatedProperty indicate that a particular feature is preserved for backward-compatibility purposes, but these may be phased out in the future. However, despite other OWL constructors, the six tags devoted to ontology versioning do not have any formal semantics expressed in description logics, so reasoning abilities are restricted with these constructors. This confirms that the novelty of the problem related to ontology changes management, combined
with the **evolution of Web data**, requires the Semantic Web community to integrate a better method to express changes in ontologies.

**Dynamic OWL (dOWL)**

Through the proposition of dOWL, the authors (Avery & Yearwood, 2003) work to correct what OWL lacks, by enriching the language with new constructors for OWL ontology evolution. It is clear that OWL provides little concerning ontology evolution as it provides no means to express: renaming a class or a property; removing a class or a property; redefining the restriction of a class; redefining the domain and the range of a property; redefining a property as being symmetric, transitive, functional or inverse functional; coalescing many classes or properties into one; or dividing one class or property into many. The dOWL language is an enhanced versioning mechanism for OWL that caters to the evolution of ontology. The dOWL language is defined as an OWL ontology (URL http://www.ballarat.edu.au/javery/2003/4/dowl#) and consists of a set of one or more OWL ontologies, followed by a set of zero or more versions that provide a mapping or translation between the old version(s) and the new version.

A version in a dOWL ontology has the same structure as an OWL ontology except that within the ontology header there are one or more dowl:importsAll or dowl:importsNone elements. The dowl:importsAll element imports all the elements of an ontology just as the owl:imports element, except the elements explicitly mentioned within the dowl:importsAll tag. The dowl:importsNone element imports none of the elements of an ontology except those elements explicitly mentioned within the dowl:importsNone tag. Other tags proposed by the dOWL language are devoted to correct the other features (listed above) that OWL lacks. Although improving some OWL drawbacks, in terms of ontology change management, dOWL does not have a formal semantics which prevents some advanced reasoning mechanisms.

**Future Trends**

Although Web data is evolving by nature, the Web is also evolving in its philosophy. According to Tim O’Reilly, the Web is moving towards an era of “data lock-in” (O’Reilly, 2004). If one takes a closer look at the WIS applications like Google mail, flickr or even Amazon, one will observe that their success is the consequence of high storage capacity and their ability to rapidly retrieve, using content managers and metadata based approaches, the data offered by these applications. For the hobby Web, the consequence is that its data is “locked-in” another company’s database. These data and associated services are then used to provide personalized data access or personalized data presentation.

Web 2.0 and future version of the Web will need to deal with data lock-in and introduce new Web applications in which the user has a priority position which in turn will give place to new kinds of evolution like “on demand” evolution where users will be the initiators of the changes and not the applications anymore. For example the subscription of users to news diffusion service using RSS stream. Technological improvements that traditionally go with Web evolution provide not only a powerful access to information but also a smarter access. This means that information is more pertinent with respect to user needs, but also that information is much better presented. These particular aspects will benefit from the acceptance of technologies like **AHS**. The retrieval of pertinent information will be done in large part with “intelligent” software able to reason on Web data thanks to Semantic Web concepts.
Nevertheless, these concepts and mainly languages for ontology evolution presented in this paper will require significant improvement to manage evolution in an appropriate way. Since ontologies are used to model a given domain which is supposed to evolve over time (see Figure 1), the current state-of-the-art on ontology evolution presented in this section only provide very little to support this evolution automatically. In fact, the ontology needs a partial (or total) reconstruction each time a change in the domain occurs. This is particularly annoying for domains that evolve in a cyclic manner. Evolution can also lead to the exclusion of a domain’s concept when it is no longer used or in the contrary, the emergence of a new concept in the domain (like for instance the concept ontology in the artificial intelligence domain in 1993).

More generally, evolution tends to modify the relations between concepts of a given either to strengthen or to weaken depending on their usage. Measurements of these modifications are important and should be derivable from the ontology. This is especially important for ontologies used for information retrieval. Actually, if one can select concepts of a given ontology with a short semantic distance, one will be able to build queries that will act as better filters to skim the result of a search.

This is also true if the domain of interest represents user behaviour. Becoming more experienced with the Web and search applications, users modify their habits when searching the Web. Ontologies that support user queries have to reflect these evolutions. Therefore, if the different kinds of domain evolutions are clearly and formally defined, it will be possible to extend the existing languages and OWL in particular with new operators to allow ontologies developed with these languages to evolve automatically (or at least semi-automatically) over time.

Some of the new proposed operators should at least have the following abilities:

1. Allow the definition of the validity duration of an assertion. For instance, how long the assertion “France has the lead of the European Union” is valid. This property would be very important to describe events that occur in a cyclic loop. This is all the most so true in the context of the Web. In fact, as illustrated in figure 1, this may occur in a data update situation. Many Web pages contains the information “last modification” or “last update” with the corresponding date. In consequence, if the author of the Web document is able to give the period for which the documents’ content is valid, it will on one hand enhance the quality regarding data relevance of the Web site information. On the other hand it will reinforce the popularity of the Web site which is important for business WIS applications.

2. Allow to specify when a concept has been added to or removed from the ontology. This is important to ensure the compatibility between ontologies which in turn will preserve basic properties for reasoning purpose which is one of the main objectives of the Semantic Web vision.

3. Allow the measurement or the specification of the semantic distance between concepts of the ontology. As explained before this will be helpful for query construction but it will also enhance the precision of the domain that is represented by the ontology. Moreover, this semantic distance should evolve dynamically over time according to the usage of the concepts or the relations between concepts in the domain. These metrics will therefore improve the quality of the ontology.

4. Introduce “change resistance coefficients” in order to manage the wanted equilibrium between change and stability of the ontology over time. For example, the concept ontology has been identified as a very important concept in the Semantic Web domain. Therefore, it
must not be excluded from the ontology (or at least in a long period of time) so this concept should have a strong coefficient that will allow to resist to changes that could try to suppress it. This is important in the context of the Web since Web site designers publish on their pages many major concepts of a particular domain. Therefore, using the appropriate metadata, designers should be able to specify stable or volatile information.

5. Allow the definition of degree of freedom that would indicate how concepts will evolve ones compared to the others. This property would strongly be correlated with the usage of the concepts of the ontology and will prevent to have ambiguity between concepts of the ontology.

Furthermore, the evolution of the ontology should follow some type of dynamic and static “well formedness” rules that have to be introduced in order to manage coherently the different automatic (or semi-automatic) modification of an ontology. If such rules are integrated in the languages, we believe that it will be much easier for metadata to be maintained over time mainly when changes in the domain take place.
If one considers the key quality aspects for Information Systems applications gathered under the ISO-9126 norm (performance, maintainability, usability, portability, security issues, etc.), the use of metadata as concepts for improving these criteria will be evident. In order to better illustrate the impact of metadata on WIS applications, we will illustrate a basic application example. Consider an online travel agency that sells travel related products and services to customers on behalf of third party travel suppliers. First, an important criterion of WIS applications is the capability of integrating several heterogeneous data coming from different Web sources and from IS’s own repositories. The information contained in a WIS application can be harvested through the Web by integrating the Semantic Web or metadata embedded in (X)HTML documents (Yahaya, Gin, & Choon, 2005). The WIS application can also be built, for historical needs, upon heterogeneous components. For instance, part of the data can be contained in a traditional relational database and the other part in a database utilizing XML. Therefore, the use of a richer metadata represented in ontology can be used to harmonize the combined use of both databases (Kashyap & Sheth, 1998). From a maintenance point-of-view, our travel agency will need to keep the information provided to Web users up to date. Should there be changes in fees metadata, reactivity can be used to detect and propagate these changes for users.

If we base the maintainability of WIS applications on metadata, should the metadata become outdated it will strongly impact the WIS application. As a consequence, we believe that the languages and techniques, if extended as proposed, are good candidates to manage ontology evolution. WIS applications will then benefit from this metadata evolution by having an up-to-date metadata for describing new content.

The availability of the application is also important for success. Our example travel agency could, to prevent failures, duplicates its data in several databases and use metadata to detect the aliveness of the server that hosts the database and react by redirecting requests to available and adequate servers (Di Marzo Serugendo, Fitzgerald, Romanovsky, & Guelfi, 2007).

This small example gives a good illustration of the value of integration of metadata in WIS applications. All that is needed at the time of design is the rigorous definition of specific metadata that will be used within future WIS applications, according to the application requirements. For instance, if security has to be the most important quality for the WIS application, metadata, like “aliveness of a resource,” “availability of a data,” etc., has to be defined. The use of metadata will improve the quality of the design of WIS applications and it will help in establishing rules at the time of design for checking the consistency of the WIS application (Stuckenschmidt & van Harmelen, 2004).

**Metadata and Adaptation to Users**

Adaptation of content to user specificities is the last point that plays an important part in the development of future WIS applications. WIS application developers focus on the presentation of data using AHS technologies (Houben, 2004; Jacquiot, Bourda, Popineau, Delteil, & Reynaud, 2006). We have presented the usefulness of metadata in the adaptation of WIS applications in regard to data evolution. Due to technological improvements that can be hardware or software based, users have a growing number of options for consulting WIS content. There are many ways to present data to users according to their own user specificities.
(experience, background, etc.), and also according to the technologies used to reach WIS content. In the case of our travel agency example, the application should know what kind of device is in use to display the most appropriate information on the user’s screen. This feature is all the more important in the case of emerging e-health Web applications. The application adapts to the user’s device and to the particular characteristics of the user. If the user is a patient, the application should display only the user information related to his/her own medical treatment, and if the user is allowed to modify special data the application must control data access. All these described features can be facilitated with the use of control metadata that can be integrated into the WIS application at the time of design. The integration of users in WIS applications will be possible only if the user is modelled in an adequate way. We believe the use of ontology can be of real value to this end. Nevertheless, improvements are needed to precisely model users or categories of users and to make the model adaptable to users and to domain evolution.
CONCLUSION

Although it is unavoidable, the evolution of Web content has not sufficiently been taken into account in the development of Web applications (in general) and in WIS applications (in particular). The emergence of new paradigms, like the forthcoming Web 2.0 and the Semantic Web, confirms that it will be impossible to continue to ignore this evolution, especially regarding WIS applications. Therefore, the use of metadata, though in its beginning stage, seems to be the most promising means in order to manage Web content evolution and improve WIS applications. Even though description of metadata through the use of ontologies has been accomplished, the existing languages and techniques studied in this chapter for ontology evolution provide only an overview of how to cope with Web content evolution. In languages, the notion of ontology versioning has been integrated but, the overwhelming number of concepts (i.e. metadata) for preserving compatibility between several versions of the ontology adds to the complexity and impact on the consistency of these structures. Furthermore, this notion is not dealt with sufficiently in the existing methodologies for ontology evolution. Ideally, ontologies have to evolve smoothly, semi-automatically at the same speed with Web knowledge evolution while remaining compatible with previous version of the ontology.

Developers of WIS applications must also have in mind another kind of evolution: knowledge of the user for whom the applications are devoted to. Many e-commerce applications draw information from the user’s profile and behaviour to improve business by proposing products that might be of interest. Companies that base their business on WIS applications need to combine both domain and user evolution. This is why the existing languages and techniques for ontology evolution need to be enriched for optimum outcome.
REFERENCES


Figure Captions

*Figure 1.* An overview of Web evolution.
I'm in Luxembourg.

```xml
<xml version="1.0" encoding="utf-8">
<root>
  <User>
    <name>John Doe</name>
    <email>john.doe@example.com</email>
    <profile>Developer</profile>
  </User>
</root>
```

Hello!

```html
<html>
  <head>
    <title>Hello from Luxembourg</title>
  </head>
  <body>
    <h1>Hello from Luxembourg!</h1>
  </body>
</html>
```
Key Terms

Hidden Web: The hidden web (or invisible web or deep web) is the name given to pages on the World Wide Web that are not indexed by search engines. It consists of pages which are not linked to by other pages, such as Dynamic Web pages based on responses to database queries. The Deep Web also includes sites that require registration or otherwise limit access to their pages.

Ontology: An ontology is an explicit and formal specification of a conceptualization. In general, an ontology describes formally a domain of discourse. Typically, an ontology consists of a finite list of terms and the relationships between these terms.

Semantic Web: The Semantic Web is an evolving extension of the World Wide Web in which web content can be expressed not only in natural language, but also in a form that can be understood, interpreted and used by software agents, thus permitting them to find, share and integrate information more easily.

Web 2.0: Web 2.0 is a term often applied to a perceived ongoing transition of the World Wide Web from a collection of websites to a full-fledged computing platform serving web applications to end users.

Web resource: Any one of the resources that are created during the development of a Web application, for example Web projects, HTML pages, JSP files, servlets, custom tag libraries, and archive files.

WIS: A Web Information System (WIS) is an information system that uses the Web to present data to its users.