D13 v0.1 Ontology Instantiation

SW-Portal Working Draft 03 August 2004

This version:
http://www.deri.at/research/projects/sw-portal/2004/d13/v0.1/20040803/
Editors:
Katharina Siorpaes
Kathrin Prantner

Authors:
Katharina Siorpaes
Kathrin Prantner
Anna V. Zhdanova

Table of contents

---

1. Introduction

This first draft of the ontology instantiation deliverable tries to give an overview over various methods how ontology instantiation can be done. The deliverable gives an overview of those methods. Furthermore, it takes a look at some existing Semantic Web Portals and how they do ontology instantiation and maintenance.
The last part of the deliverable draws a conclusion from the investigated aspects concerning the choice, how ontology instantiation should be done in the DERI Semantic Web Portal group.

2. Ontology Instantiation Techniques

2.1 Forms

The insertion of data can be done by using web-based forms. This methodology enables a wide audience to participate in creating instances and is kind of low-level entry for users with little knowledge.
An example for a tool which assists in providing instance forms is Knote [7] which creates the forms directly from the ontology.

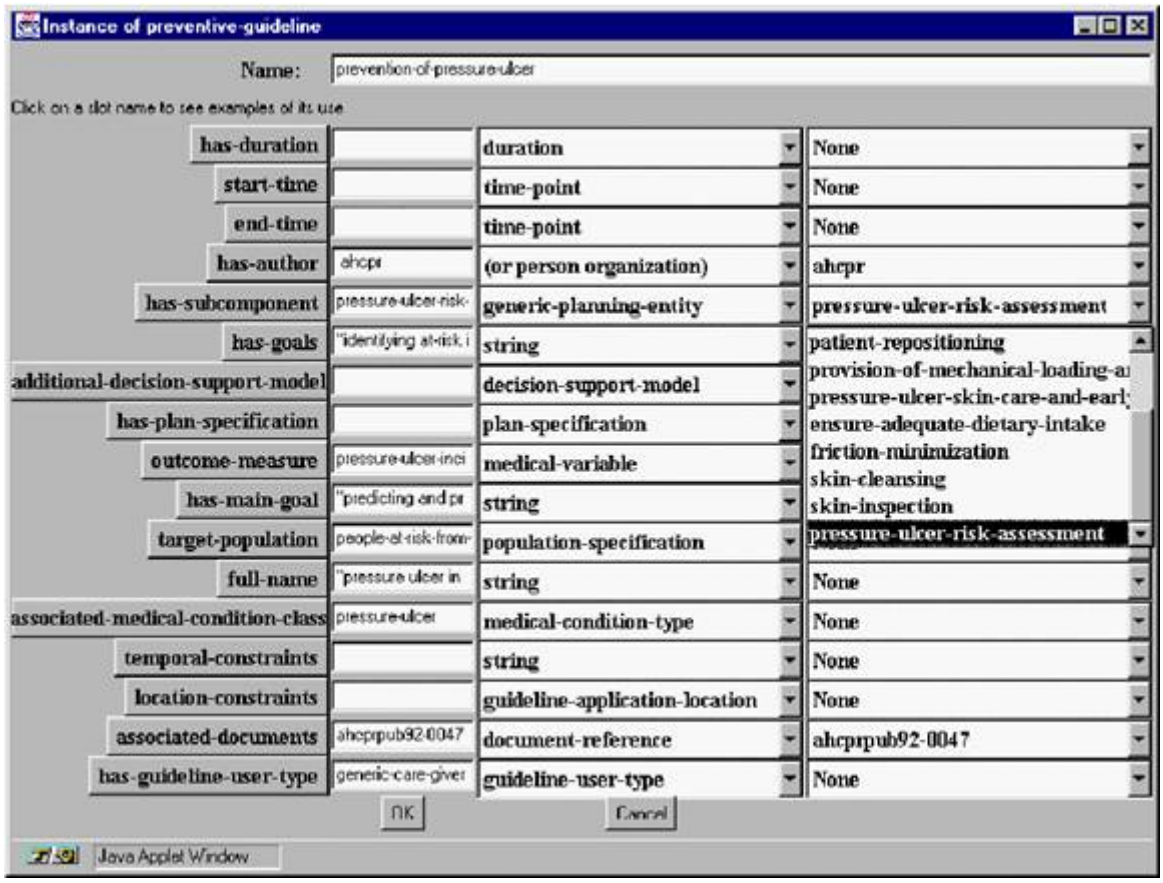

Figure 1 - Screenshot of Knote [7]

After the user has chosen a slot, the form is created dynamically.
The drawback with this method is, that all the instances have to be created and maintained manually. On the other hand it is a pretty intuitive solution and could serve as an easy to use starting point.

2.2 Wrappers

A wrapper is a shell around a source of information that makes its content accessible regardless of its implementation details. In order to deal with the weak structure Web content usually has, wrappers can rely on several techniques, such as heuristics to guess where the needed information resides, Natural Language Processing for reducing the ambiguities in the content, Information Retrieval, etc. Wrappers can be constructed in various ways ranging from manually programming to semi-automatically generation.

The basic idea behind wrapper-based extraction of data is that a wrapper, also called screen scraper, is used to extract information from external documents with a similar structure. There are two kinds of information sources: The first are semi-structured

information sources such as HTML and the second are structured information source such as databases. The wrapper would extract data which is intended to be extracted, and send it to the server converted into other formats, i.e. RDF, XML etc. When the server receives a new file, it starts another process to parse the given file and to add new information to a Knowledge Base. At this point it would become necessary to proof the content of the file or if there is a duplication of the new information. A big advantage of using wrappers is that they allow automatic, fast and efficient extraction.

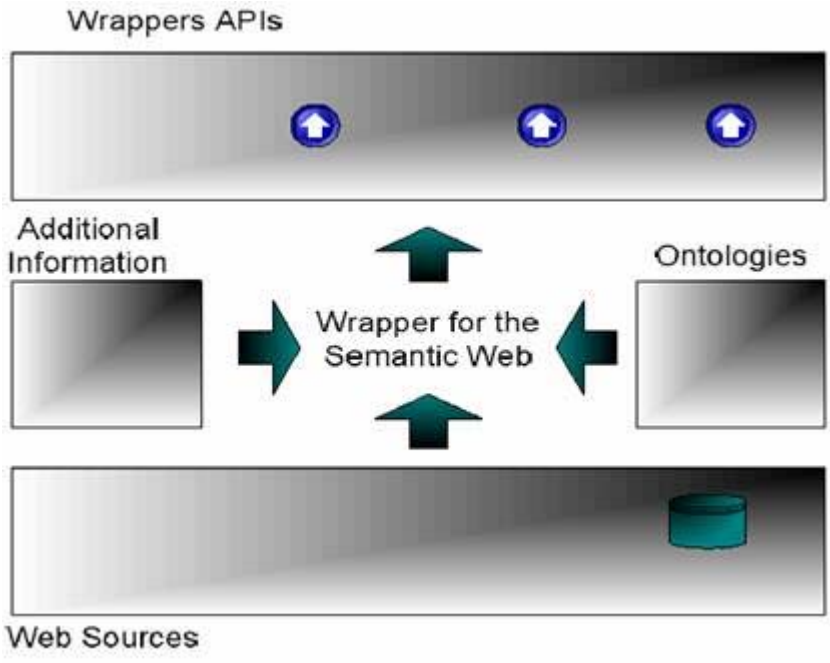

Figure 2 – Wrappers fort he Semantic Web [2]

2.3 Ontology mapping patterns

Ontology mapping is the process whereby two ontologies are semantically related at conceptual level, and the source ontology instances are transformed into the target ontology entities according to those semantic relations. Basically, there are three dimensions of ontology mapping:
First, there is discovery, which is defining the relations between ontologies either manually, automatically or semi-automatically.
Second, a language for the representation of the relations between the ontologies has to be found.
Third, the execution is performed by changing instances of source ontologies to an instance of the target ontology.

Various tools exist to support ontology mapping, such as PROMPT [10], ONION [8], Chimaera [9], GLUE [3] etc.

2.4 Natural Language Processing

One of the more promising approaches to process actual web content is to use Natural Language Processing techniques. The Web can be a huge source of information. Natural Language Processing aims to enable automatic extraction and consolidaton of knowledge from Web documents. This knowledge can be used to instantiate a given ontology, which dictates the type and form of knowledge to extract. Automatic instantiation of ontologies

and building of knowledge bases from the web corpus is very effective. NLP systems can process any natural language input (including speech) and the process ends with the pragmatic level where actions are taken according to the input.
Tools that realize Natural Language Processing divide documents gathered from the web into paragraphs and sentences. Each sentence is analysed syntactically and semantically to identify any relevant knowledge to extract. This would be a fully automatic approach of feeding the ontology with knowledge extracted from unstructured text.
An example for an interesting tool that is concerned with Natural Language Processing and automatic ontology instantiation is Artequakt [12]. In this project, information is extracted with respect to a given ontology and provided as XML or RDF file and then fed to the ontology.

3. Ontology Instantiation in various Semantic Web Portals

To get an overview about how the state-of-the-art procedure looks like concerning ontology instantiation, we investigated the following Semantic Web Portals. The following section shows how they do ontology
instantiation and how these instances are created and maintained.

3.1 Esperonto

User Roles
In the Esperonto [http://esperonto.semanticweb.org/] portal users have different permissions, which means that various user roles exist.
such as Administrators, guest users and members.
While Administrators are allowed to create, edit and browse any information on the portal, guest user can only browse the public information.
Members have access to various areas. Basically, they are allowed to create and edit information items.
Validators
Esperonto is the only portal that does not have any form of quality control: every user can add anything to the portall. There are no validators. Administrators can delete inappropriate items, but basically anything can be published.
Creating instances
To create a new item the user has to select the type of information (ontology concept) the user is about to create. After having selected an ontology concept, the user is presented with a form to enter the name of the instance and its description. Immediately a new instance is created and thus implicitly assigned to the ontology. The name the user enters when creating the information item also works as an identifier, which may cause problems, because the user does not necessarily know all the names of all the instances in the portal.
Editing instances
Depending on the user right of an instance, existing instancescan be edited later, even though the name, which works as an identifier, and description of the instance cannot be changed.
Deleting instances
Furthermore, as a member it is not possible to delete instances, even if they belong to the member who wants to delete the instances. Deletion is restricted to administrators.

3.2 Knowledgeweb Portal

The Knowledgeweb Portal [knowledgeweb.semanticweb.org] is developed by the same team as Esperonto portal. So the two portals are very similar. The Esperonto project is done and the Knowledge web portal is still active.

User Roles
While Administrators are allowed to create, edit and browse any information on the portal, guest user can only browse the public information.
Members have access to various areas. Basically, members are allowed to create and edit information items.
Validators
There are no validators. Administrators can delete inappropriate items, but basically anything can be published.
Creating instances
To create a new item the user has to select the type of information (ontology concept) the user is about to create. After having selected an ontology concept, the user is presented with a form to enter the name of the instance and its description. Immediately a new instance is created and thus implicitly assigned to the ontology. The name the user enters when creating the information item also works as an identifier, which may cause problems, because the user does not necessarily know all the names of all the instances in the portal.
Editing instances
Depending on the user right of an instance, existing instances can be edited later, even though the name, which works as an identifier, and description of the instance cannot be changed.
Deleting instances
Furthermore, as a member it is not possible to delete instances, even if they belong to the member who wants to delete the instances. Deletion is restricted to administrators.

3.3 Mondeca

User Roles
Mondeca implements three kinds of users: administrators, validators and members. Administrators have unlimited rights to write, read and delete in the portal. Validators are responsible for quality control on the portal. They decide whether an item can be published or not. Each member is assigned a certain workspace and can create instances.
Validators
Validators decide whether an item of information can be published or not. Basically, there are two states an information item can have: “proposal” or “validated”, which follows the same line as many other Semantic Web portals except Esperonto do – an instance of a concept has to be validated before it can be published on the portal. The drawback here is, that a validator does not get a notification about a new information item waiting to be validated.
Creating instances
In the Mondeca portal [http://www.mondeca.com] instances of ontology concepts can be added in various ways: it can be added by the end user via forms, automatically using linguistic tools or by extracting information from structured or semi-structured sources.
To fill the knowledge base with data initially, this can be done by importing OWL files using XTM as an export format.
The user can create instances by using a form. The type of form depends on the concept that the user wants to create, who provides the necessary information to create the ontology instance.
The information item is implicitly assigned to ontology. Additionally, it can be assigned to one or more keywords in a defined thesaurus.
Furthermore, it is possible to establish relations among items of information.
Editing instances
When editing an instance, values and semantic associations have to be changed separately. A big drawback is that there is no user interfaces to perform this functionality, so external tools, such as Protégé have to be used.
Deleting instances

Furthermore, the user can determine an expiration date to define for how long the information should be on the portal. This is also how deletion of items works. Moreover, one can also delete an item by using the user interface - depending on the right a user has. When an information item is deleted, all the semantic associations are deleted, too.

What is special about Mondeca is that there is a functionality to extract the author's name automatically out of a WORD document.

3.4 OntoWeb

User Roles
In this portals, there are two different user roles: "normal" users and administrators. While administrators can publish, reject, retract, delete and change all information items, normal user can only edit their own information items.
Validators
In the OntoWeb portal there are validators, the administrators, who are responsible for the quality. In the OntoWeb portal there is a distinction between private and public information items. Private items are only visible to the creator and the administrator, while the public ones can be viewed by every user once it has been reviewed by a portal reviewer.
Creating instances
The OntoWeb portal [http://www.ontoweb.org]differentiates between several types of information items defined in the ontology. For each
of these types there is a form, which is used to create new instances. This form is divided in three sections of
information: a section called "base data" containing a short description; furthermore another section containing the
values for the attributes of the ontology concept, which includes an upload functionality; finally, a section called "meta data" containing instructions on how the portal should handle the new item. Public items also include an expiration date until which it will be accessible. When a user creates a new information item, it is automatically immediately assigned as an instance of the ontology.
Editing instances
Once created, each information item has its individual editing options. Users who have the appropriate rights can edit instances to a certain extend: as soon as an item has been published or submitted, the user can only rename the item or change its status, but not change its declarative description.
Deleting instances
Users are allowed to delete their instances independent from the status of the information items.

3.5 K42

User Roles
In the K42 portal [http://www.research.ibm.com/k42/]users are only allowed to browse through existing information while the administrator is responsible for creation, publication and maintenance, which means that all information items are provided and maintained by the portal administrator.
Validators
The Administrator who is responsible for everything on the portal is also performing quality control.
Creating instances
A tool named WebAuthor allows web-based editing of topic maps. A tool named Ontogen can be used to create and edit information items based on web forms. Still, it is not possible to upload documents associated with the instance. This has to be done

separately.
The information items are published immediately which means that they are published when they are submitted to the system via the creation opportunities. So the border and between the creation and the publication phase is intangible.

Editing instances

The administrator is able to edit all the information items on the portal.

Deleting instances

The administrator is able to delete all the information items on the portal.

	Esperonto	KnowledgeWeb	Mondeca	OntoWeb	K42
User Roles	Administrator, User, Guest	Administrator, user, guest	Administrator, validators, users	Administrator, users	Administrator, users
Validators	no	no	Validators	Administrator	Administrator
Creating instances	forms	forms	forms, wrapping possible	forms	forms, only administrator
Editing instances	possible. Name as an identifier must not be changed	possible	possible, but there is no user interface; tools must be used (e.g. Protege)	possible. Declarative description must not be changed.	only possible for the administrator
Deleting instances	only administrator	only administrator	depending on the expiration date	possible for users and administrator	only possible for the administrator

4. Conclusion

After having investigated some Semantic Web Portals and shortly summarized the possibilities of instantiating the DERI Semantic Web Portal ontology, it is obvious that there are various approaches.
Ontology mapping patterns seem to be no adequate solution for this project at the moment because there are no fitting resources. Maybe this is an opportunity to consider when the ontology is populated and has been used for a longer period of time.
Using forms seems to be a very appealing possibility.
It is a very straight forward approach and seems to be a good solution as a starting point. It is an easy to use technology, so that users do not need expertise. The big drawback is that it is a very static approach. The instances have to be maintained manually and also the creation process is quite time-consuming.
The alternative is to consider using wrappers to gather information dynamically from the

web pages of portal members. Setting up a procedure using screen scrapers might be more complex but in the end it pays, because it is much easier to maintain. The main problem with wrappers is that we do not have web pages that are structured well enough and that contain enough information.
How time consuming and complex the implementation of both, wrappers and forms, is depends on the complexity.
The deri.at and deri.ie member pages could serve as a starting point to use a wrapper approach because these pages are well structured. Even though they contain very little information about each member, such as name, telephone number, e-mail adress, etc., this information could be extracted and fed to the ontology. This approach seems to be tangible and fast to realize. It seems to be a good way to get things running, especially considering the prototype of the Semantic Web Portal group: the prototype is collecting data from the ontology to create websites automatically.
Probably a good trade off would be to use a form based approach later to feed the ontology with more detailed information.

5. References

[1] Alani, H., Kim, S., Millard, D., Weal, M., Hall,W., Lewis,P., Shadbolt,N., 2004: *Using protégé for Automatic Ontology Instantiation,* http://eprints.ecs.soton.ac.uk/archive/00009479/01/Alani.pdf, 2004-08-01

[2] Contreras,J., Benjamins,R., Martin,F., Navarrete,B., 2003: *Esperonto Project: D31 Annotation Tools and Services,* http://esperonto.semanticweb.org/semanticportal/jsp/ViewInstance.jsp?term_name=Deliverable&instance_name=D3.1%3A+State+of+the+art+on+annotation+tools+and+services&order=Deliverable&submit=View 2004-08-02

[3] Doan, A., Madhavan, J., Domingos, P., Halevy, A., 2002: *Learning to map between ontologies on the semantic web.* In The Eleventh International WWW Conference , Hawaii , US , 2002.

[4] Engels,R., Fensel,D., Van Harmelen,F., Iosif,V., Kampman,A., Krohn,U., Reimer,U., Studer,R., Sure,Y., 1999: *Content-driven Knowledge Management through Evolving Ontologies*, http://www.ontoknowledge.org/downl/del33.pdf , 2004-08-01

[5] Fensel,D., Bussler,C., Ding,Y., Kartseva,V., Klein,M., Korotkiy,M., Omelayenko,B., Siebes,R., 2002: *Semantic Web Application Areas,* http://www.cs.vu.nl/~mcaklein/papers/NLDB02.pdf, 2004-07-20

[6] Lausen,H., Stollberg,M:Lara,R.;Ding,Y., Han,S., Fensel,D., 2004: *Semantic Web Portals – State of the Art Survey*, http://deri.ie/publications/techpapers/documents/DERI-TR-2004-04-03.pdf, 2004-08-01

[7] Motta, E., Buckingham Shum, S., Domingue, J., 2000: *Ontology Driven Document Enrichment,* http://kmi.open.ac.uk/projects/marchmont/OntoDocEnrich-IJHCS-2000.pdf, 2004-07-24

[8] Mitra, P., Wiederhold, G., Kersten, M., 2000: *A graph-oriented model for articulation of ontology interdependencies.* In Proceedings Conference on Extending Database Technology 2000 (EDBT'2000) , Konstanz , Germany , 2000.

[9] McGuinness, D.L., Fikes, R., Rice, J., Wilder, S., 2000: *An environment for merging and testing large ontologies.* In A. G. Cohn, F. Giunchiglia, and B. Selman, editors, Principles of Knowledge Representation and Reasoning: Proceedings of the Seventh International Conference (KR2000) . Morgan Kaufmann Publishers, San Francisco , CA , 2000.

[10] Noy, N.F., Musen, M.A., 2000: *PROMPT: Algorithm and tool for automated ontology merging and alignment.* In Seventeenth National Conference on Artificial Intelligence (AAAI-2000) , Austin , TX , 2000.

[11] Staab,S., Angele,J., Decker,S., Erdmann,M., Hotho,A., Maedche,A., Schnurr,H.P., Studer,R., Sure,Y., 2000: *Semantic Community Web Portals,* http://www9.org/w9cdrom/134/134.html, 2004-07-25

[12] Weal, M.J., Alani, H., Kim, S., Millard, D.E., Hall, W., Lewis, P.H., Shadbolt, N., 2003: *Web based Knowledge Extraction and Consolidation for Automatic Ontology Instantiation*, http://eprints.ecs.soton.ac.uk/archive/00008325/01/Alani-SEMANNOT-camera-ready.pdf, 2004-08-03

Acknowledgement

The editors would like to thank to all the members of the SW Portal working group for their advice and input to this document.

$Date: 2004/08/02 21:55:41 $

webmaster