



OMWG D7.1: Requirements for Mapping and MergingTool

DERI OMWG Working Draft 6 December 2004

This version:

<http://www.omwg.org/2004/d7/d7.1/v0.1/20041206/>

Latest version:

<http://www.omwg.org/2004/d7/d7.1/v0.1/>

Previous version:

<http://www.omwg.org/2004/d7/d7.1/v0.1/20041123/>

Authors:

Fancois Scharffe

Editors:

Francois Scharffe, Atanas Kiryakov

Copyright © 2004 [DERI](#)®, All Rights Reserved. [DERI](#) liability, trademark, document use, and software licensing rules apply.

Table of contents

[1 Tool Requirements](#)

[1.1 Interoperability and compatibility](#)

[1.2 Genericity](#)

[2 Functional Requirements](#)

[2.1 Ontology Alignment Language Requirements](#)

[2.2 Mapping Patterns](#)

[2.3 Mapping algorithms programming interface](#)

[2.4 Ontologies merging function](#)

[2.5 Graphical user interface](#)

[2.6 Runtime module](#)

[2.6.1 Query rewriting](#)

[2.6.2 Instances transformation](#)

[2.7 Link to the repository](#)

[3 Interface Requirements](#)

[3.1 Version Browser](#)

[3.2 Transformation Editor](#)

[3.3 Notification Agent](#)

[4 Implementation Priority List](#)

[5 global architecture](#)

[6 Conclusion](#)

[7 Acknowledgement](#)

[Appendix A. References](#)

1 Tool Requirements

1.1 Interoperability and compatibility

The ontology mapping tool will be part of the Ontology Management System, it then have to communicate the other modules of the system.. More importantly, it can benefit from the features offered by other tools in the ontology management suite. With this requirement this tool contributes to the overall interoperability and compatibility requirement for the tool suite. [Deliverable 4, Section 2](#) situates the versioning tool in the OMS framework.

This tool will interoperate with the versioning tool, to store and retrieve the mappings and the merged ontologies, and of course to work with different version of ontologies to mediate.

A runtime module will also be develloped, following the requirements specified in the SWWS project, deliverable 5.2.

1.2 Genericity

The mediation should be able to work with different ontologies languages. A generic ontology language conversion tool wich mediate between different ontology languages will be a general tool of the OMS. This tool will be elaborated on the basis of ORDl.

This module is dealing with mapping algorithm. As these algorithms are currently fast evolving and research on the is very active, we expect the mediation tool to include an API supporting the implementation of different algorithms. With this possibility, the mediation tool will be able to follow the evolutions of the research area.

2 Functional Requirements

In this section we elaborate on the functional requirements from a research point of view. For a detailed implementation priority list with respect to these requirements we refer to Section 4.

2.1 Ontology Alignment Language Requirements

Core Requirements for a powerful mapping-language are:

- The basis is a powerful rule language.
- The rule language needs a rich set of (possibly extendable) built-in predicates.
- The rule language needs aggregate functions.

The mapping language consists amongst others of aggregate functions and built-in predicates.

2.2 Mapping Patterns

Patterns are templates that match the more usual mistakes between two ontologies. The use of predefined patterns considerably reduce the mapping designer task. In this solution we propose the use of a pattern language to define them, a pattern library allowing storing and retrieving them efficiently.

2.3 Mapping Algorithms Programming Interface

The architecture of the module allows the use of different mapping algorithms. These algorithms are stored and can be combined to create efficient mappings. The interface specifies the ontology language in input and the mapping language in output. This will be provide support for the evolutions of the research in ontology mediation.

2.4 Ontologies merging function

This fuction use a mapping to merge two ontologies schemas and instances. It will include parameters to specify the mapping to use and the range of the merging (wether the instances are also merged or not)

2.5 Graphical user interface

This interface plays the main role in the mapping module. It allows the user to graphically create or modify mappings by linking similar entities. Mapping proposals as results of the mapping algorithms are also integrated in this part of the component. We will detail the GUI in [chapter 3](#) of this deliverable.

2.6 Runtime module

This module is used by the reasoning part of the ontology management system. It can also be implemented as a web service but we won't discuss this here.

This module uses the mappings to perform the following tasks

2.6.1 Query rewriting

Used to rewrite a query written for an ontology into one for another ontology. This process uses the mapping between the two ontologies or proposes to create one using the mapping module.

2.6.2 Instances transformation

Use to transform instances from one ontology to another. This process also uses the mapping between the two ontologies.

2.7 Link to the repository

Link to the ontology repository (ORDI) to store and retrieve mappings, mapping patterns and merged ontologies.

3 Interface Requirements

3.1 Manual Mapping Interface

A graphical tool to realize manual mappings between two ontologies. The interface allows the user to select entities from the ontologies and to apply a pattern on them. This interface uses some modules of the Ontology Editing and Browsing tool (see [D8.1](#)).

3.2 Pattern Library

This interface provides pattern management tools, such as editing and browsing of patterns, storage and retrieval of pattern.

3.3 Matching algorithms Library

This interface provides a list of matching algorithms. The user can select an algorithm and two ontologies and execute the algorithm. the user may then validate the proposals of the algorithm.

4 Implementation Priority List

For the implementation we have distinguished three phases:

- Version 1: From M13 to M18
- Version 2: M19 to M24
- Version 3: after M24.

The Vs indicate in which phase which requirement is being initially tackled.

Req. ID.	Versioning Requirement	Version 1	Version 2	Version 3	Priority
V1	Interoperability/Compatibility	V			(affects all implementation)
V2	Genericity	V			(affects all implementation)
V3	Mapping Language	V			HIGHEST
V4	Mapping Patterns	V			HIGHEST
V5	Manual Mapping Interface	V			HIGHEST
V6	Merging function	V			HIGHEST
V7	Links to the repository		V		HIGH
V8	Mapping Algorithms API		V		HIGH
V9	Runtime Module			V	HIGH
V10	Mapping Algorithms Research			V	LOW

5 global architecture

Figure 1: Mediation Tool Architecture

Figure 2: Mediation Tool Architecture



Figure 3: Mediation Tool Architecture



6 Conclusion

The mediation component will in a close future integrate a mapping language and a mapping patterns management tool as part of a manual graphical mapping environment. It will also include a programming interface, allowing to plug different mapping algorithms, this making the mediation component able to follow the evolution of research in this area. This component will be linked with others parts, the whole being a homogeneous Ontology Management System.

7 Acknowledgement

The work is funded by the European Commission under the projects DIP, Knowledge Web, Ontoweb, SEKT, SWWS, Esperanto and h-TechSight; by Science Foundation Ireland under the DERI-Lion project; and by the Vienna city government under the CoOperate programme.

The authors would like to thank to all the [members of the OMWG working group](#) for their advices and inputs to this document.



\$Date: 2004/10/22 16:12:55 \$

[webmaster](#)