A framework to specify, extract and manage Topic Maps driven by ontology

Giovani Rubert Librelotto
UNIFRA - Centro Universitário Franciscano
Rua dos Andradas, 1614
Santa Maria, RS, Brazil, 97010-310
+55 55 9129 4080
giovani@unifra.br

José Carlos Ramalho
University of Minho
Department of Informatics
Campus de Gualtar
Braga, Portugal, 4710-057
+351 253 604460
jcr@di.uminho.pt

Pedro Rangel Henriques
University of Minho
Department of Informatics
Campus de Gualtar
Braga, Portugal, 4710-057
+351 253 604460
prh@di.uminho.pt

ABSTRACT
The ability to extract and merge data that from documents (or databases) of different types, in order to acquire knowledge from a vast repository of information, is of unquestionable value. However that desirable integration is not an easy task. Different approaches can be followed to achieve it, ranging from the merge of resources (implying their conversion to a common format) till the fusion of the extracted parts. The idea is to interoperate those resources keeping them independent, without changes or transformations, creating over them an integration layer that gives us a general overview, as the information slices were gathered. This is possible creating a semantic network, or a conceptual map, over the resources, which relates data items among them mapping each one to its different occurrences in the repository; formally speaking, that conceptual map corresponds to the ontology that describes the knowledge we want to acquire. In this paper, we introduce Metamorphosis, a Topic Maps oriented environment to extract data from heterogeneous information repositories and to generate a browser and conceptual navigator for the extracted knowledge.

Categories and Subject Descriptors
H.3.1 [**Information Storage and Retrieval**] Content Analysis and Indexing – *abstracting methods*, *dictionaries*, *indexing methods*, *linguistic processing*, *thesauruses*.

General Terms
Algorithms, Management, Documentation, Reliability, Experimentation, Standardization, Languages.

Keywords
Topic Maps, Ontologies, Information Systems, Interoperability, Semantic Web.

1.INTRODUCTION
Daily, a lot of data is produced by every institution or company. To satisfy the storage requirements, these organizations use most of the times relational databases, which are quite efficient to save and to manipulate structured data. Unstructured data (appearing inside documents) is stored in plain or annotated text files.

There is a problem when these organizations require an integrated view of their heterogeneous information systems. It is necessary to query/exploit every data source, but the access to each information system is different. In this situation, there is a need for an approach that extracts the information from those resources and fuses it. Usually this is achieved either by extracting data and loading it into a central repository that does the integration before analysis, or by merging the information extracted separately from each resource into a central knowledge base.

Topic Maps [12] are a good solution to organize concepts, and the relationships between those concepts, because they follow a standard notation – ISO/IEC 13250 [2] – for interchangeable knowledge representation. We are using successfully, for some years, this technology for classification and integration of documents in the area of digital archiving.

However, the process of ontology development based on topic maps is complex, time consuming, and it requires a lot of human and financial resources, because they can have a lot of topics and associations, as well as the number of resources can be very large.

To overcome this problem, we developed Metamorphosis. Metamorphosis makes possible the Topic Maps extraction, validation, storage, and browsing. It is composed of three main modules: (1) Oveia extracts data, from heterogeneous information systems, according to an ontology specification, and stores it in a topic map; (2) XTche validates the generated topic map, according to a constraint specification; (3) Ulisses browses the topic map, enabling a conceptual navigation and query over the resources.

This paper describes the integration of heterogeneous information systems using the ontology paradigm, in order to generate a homogeneous view of these resources. The remainder of the paper is structured in the following sections: in section (sec.2) will introduce Metamorphosis, and then a description of each module is presented with some detail (Oveia in sec.3, XTche in sec.4 and Ulisses in sec.5). Before concluding remarks (sec.7) we compare our proposal with related work (sec.6).

SIGDOC'08, September 22–24, 2008, Lisbon, Portugal.

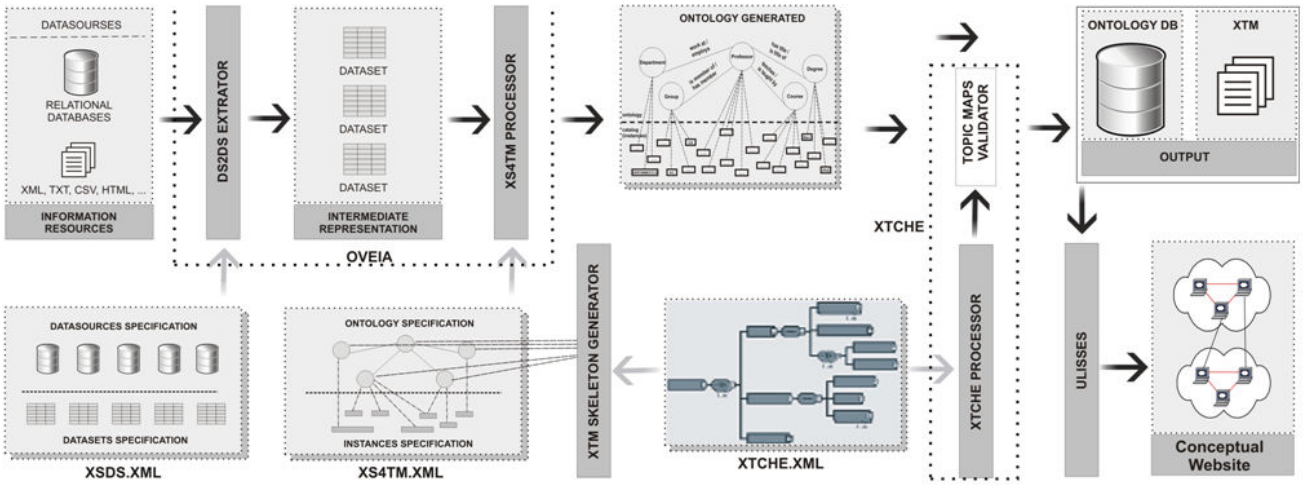

Figure 1: Metamorphosis Architecture

2.METAMORPHOSIS: AN ENVIRONMENT TO DEAL WITH TOPIC MAPS

Topic Maps are very well suited to represent ontologies [17]. Ontologies play a key role in many real-world knowledge representation applications, and namely the development of Semantic Web. Ontology is a way of describing a shared common understanding, about the kind of objects and relationships which are being talked about, so that communication can happen between people and application systems [6]. In other words, it is the terminology of a domain (it defines the universe of discourse). As a real example consider the thesaurus used to search in a set of similar, but independent, websites.

The ability of Topic Maps to link resources and to organize these resources according to a single ontology, will make Topic Maps a key component of the new generation of Web-aware knowledge management solutions. In addition, the growing repertoire of techniques for simplifying, merging and interrelating ontologies can be used to combine or articulate Topic Maps representing different ontologies, thus enabling different sets of information resources to be used together in a controlled and scalable way [4].

One of the first Metamorphosis' applications was the production of website maps for conceptual navigation; another of our former concerns was the contents publishing in the context of e-learning. Metamorphosis can be also used to test some functionalities of a dynamic web system because it creates, in a fast way, a web interface that interacts directly with data sources.

Figure 1 shows Metamorphosis' architecture that came up from the principles underlying our proposal. This architecture is composed of:

(1) *Information Resources*: It is composed of the data sources: XML documents, databases, Web pages, etc.

(2) *XSDS and XS4TM specifications*: They are domain specific languages to define the topic map extraction.

(3) *Oveia*: The processor that builds topic maps. Its core is a processor that extracts the topics instances from the information resources and builds a topic map. It reads and processes the XSDS and XS4TM specifications.

(4) *Generated topic map*: The topic map automatically generated by Oveia stored as an XTM file or alternatively a relational database.

(5) *XTche specification*: A topic maps constraint specification language based on TMCL (Topic Maps Constraint Language) [11] that allows to define rules for topic maps semantic validation.

(6) *XTche Processor*: The processor that consumes the previous XTM file and verifies the topic map according to a set of constraints defined in XTche language.

(7) *Valid topic map*: The previous topic map automatically validated by XTche.

(8) *Ulisses*: The processor that takes a topic map and produces a whole semantic Web site, a set of Web pages where it is possible to navigate through structural or syntactic links as well as through a network of concepts.

(9) *Conceptual Web site*: It is the generated Web site that allows the semantic navigation over the topic map extracted from information resources.

2.1.Metamorphosis Repository

Although XTM is a good format for interchange it is not so good for storage. When we refer to storage we are meaning the capability of storing a Topic Map and efficiently being able to query it. XTM is easy to process and for instance to translate it into another format. But querying XTM is complex.

The Topic Map model is not hierarchical; every relation is materialized as a reference. Gathering all the information about a topic is very complex. The obvious choice for storage is a database. For this case we had three options: an XML database [3], an Object Oriented Database [8] or a Relational Database. Since the Topic Map model does not match the XML model XML databases were discarded. Almost for the same reasons OO databases were also discarded. That left us with the relational model as the target for our storage solution.

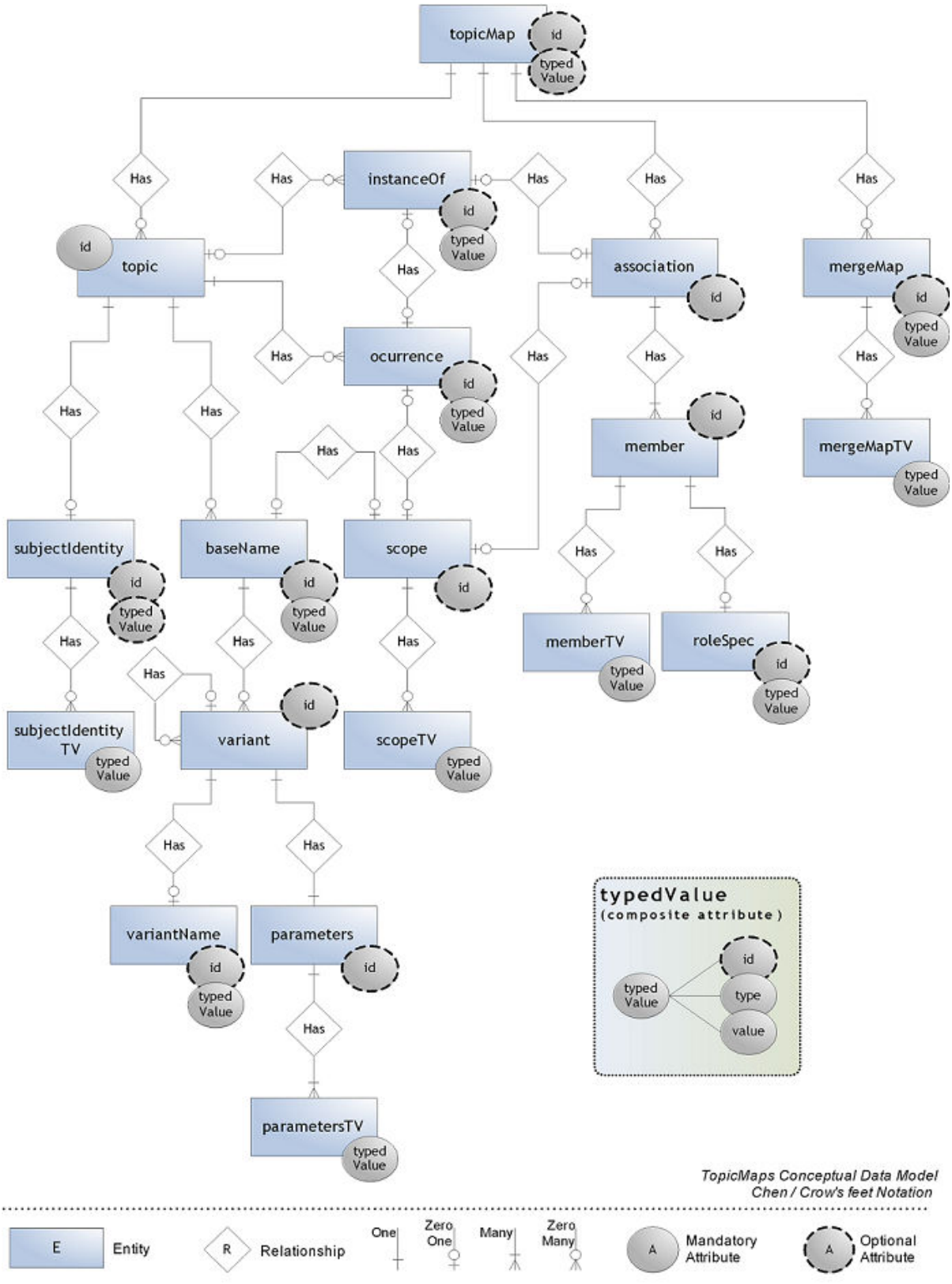

Figure 2: Relational model schema

The next step would be the specification of a Topic Map Relational Model. We have considered two approaches: look at the Topic Map Reference Model [5, 7] and derive the relational model from it or look at the XTM model and work from there. We decided to work over the XTM model and see if we could reach a model similar to the Topic Map Reference Model.

2.1.1.Data Model

This section defines an abstract model for Topic Maps which makes explicit the implicit data models of ISO 13250 and XTM 1.0.

The model is intended to present one possible approach to specifying a data and processing model for topic maps, believed by the author to be preferable to other proposed approaches. It is hoped that this model may represent a first step on the way to a complete model for topic maps.

First, we looked at the XTM model and raised the following subject list (and correspondent content model):

*topicMap = (topic|association|mergeMap)**
topic = (instanceOf|subjectIdentity|baseName|
*occurrence)**
instanceOf = (topicRef|subjectIndicatorRef)
subjectIdentity = resourceRef|(topicRef|
*subjectIndicatorRef)**
baseName = (scope?|(topicRef|subjectIndicatorRef|
resourceRef)+|baseNameString|variant)*
scope = (topicRef|subjectIndicatorRef|resourceRef)+
variant = (parameters, variantName?, variant)*
parameters = (topicRef|subjectIndicatorRef)+
variantName = (resourceRef|resourceData)
occurrence = (instanceOf?, scope?, (resourceRef|
resourceData))
scope = (topicRef|subjectIndicatorRef|
resourceRef)+
association = (instanceOf?, scope?, member+)
member = (roleSpec?, (topicRef|subjectIndicatorRef|
resourceRef))*
*mergeMap = (topicRef|subjectIndicatorRef|resourceRef)**

After some exercise with the leaf nodes of this we list end with the following types that cover any element in a topic map in Table 1.

Table 1. Types to cover any element in a topic map

(topicRef | subjectIndicatorRef | resourceRef)
(topicRef | subjectIndicatorRef)
(resourceRef | resourceData)
resourceRef
baseNameString

This result means that any Topic Map node can be represented with one of these five types. To store any of these five types we only need a triple: identifier, value and type. Consider the following example in Table 2.

This exercise enabled us to simplify the model and to reach the relational model showing in Figure 2. With this specification we have implemented a Topic Map Repository that is the core component of Metamorphosis. In the following sections we will give some details about the integration of the other components with the repository.

Table 2. Stored Values

ID	Type	Value
TR982	topicRef	#University
SIR500	subjectIndicatorRef	http://www.uminho.pt
BNS32	baseNameString	U. Minho
RD444	resourceData	UM is ...
RD446	resourceRef	http://www.uminho.pt /students

2.2.Topic Maps Discovery

Topic Map Discovery is an API that is being developed in order to work with the repository. For the moment it is composed of two parts: a topic map manager and a browser.

The topic map manager lets you upload and download topic maps in XTM syntax and delete a topic map from the repository (soon it will enable the user to edit stored topic maps).

In the next sections we are going to discuss the main pieces of this architecture: Oveia, XTche, and Ulisses, in order to demonstrate how the overall system can accomplish the task we have stated at the beginning.

3.OVEIA

For pages other than the first page, start at the top of the page, and continue in double-column format. The two columns on the last page should be as close to equal length as possible.

The ontology extractor – Oveia – is based on ISO/IEC 13250 Topic Maps [2]. Oveia extracts information fragments from heterogeneous information systems according to an XSDS specification and builds the topic map according to an ontology specified in XS4TM language.

The Oveia architecture is shown in Figure 3 and it is composed mainly of five components. The dataset extractor receives an XSDS specification – providing metadata about the physical data sources that will be used to query each source in order to get the data needed for the ontology construction – and generates the intermediate representation (called datasets) – containing the data extracted from resources. The XS4TM processor takes as input these datasets and an XS4TM specification generating a topic map, in an internal format. An output generator stores the topic map in an *OntologyDB* or in an XTM file. The following subsections describe this architecture in detail.

3.1.XSDS – XML Specification for Data Sources

Oveia supports the concept of extraction drivers. A driver extracts data from a data source and stores it in an intermediate representation, called datasets. XSDS language defines the transformations and filters over the data sources. XSDS gives precise information about each data source that should be scanned to extract topics and associations.

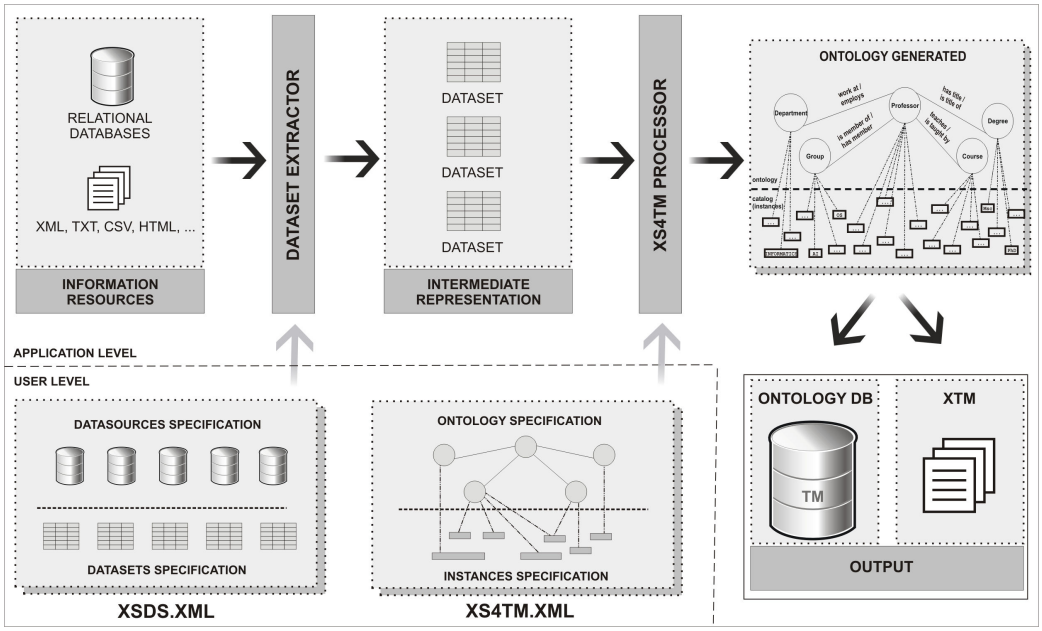

Figure 3: Oveia Architecture

An XSDS specification has two parts: *datasources* and *datasets*. The first one defines the path to the physical resources. This part has a set of attributes that indicates which extraction driver will be used and provides values for the corresponding parameters. The last one declares which data (record fields or DTD elements) must be extracted from each datasource. A datasource can be used to specify the extraction of several datasets.

3.2.Datasets: Intermediate Representation

The datasets compose the intermediate representation that contains the extracted data from the resources. Each dataset has a relation to an entity in these resources and it is represented through a table, where each line is a record following the structure specified in XSDS. The datasets representation guarantees that Oveia sees a uniform data structure that represents all the participating resources.

The dataset declaration is composed by a query to extract the data from the resources. Each dataset has a unique identifier that will be used throughout the architecture to reference a particular dataset.

The datasets are very simple, while providing the expressive power and flexibility needed for integrating information from disparate sources.

The Dataset Extractor[6] is composed of several extraction drivers (at moment, two), each one responsible for handling a specific type of source.

The driver uses the appropriate technology to make the connection (e.g. JDBC – Java DataBase Connectivity – for databases, and an XML parser for annotated documents), and then the extraction of data is expressed in the query language adequate to the type of source in use: SQL will be used to extract information from a relational database while XPath will be used for the extraction in XML documents. Finally, the extracted data is stored in the datasets.

3.3.XS4TM — XML Specification for Topic Maps

XS4TM is a domain specific language conceived to specify the process of ontology extraction from information systems; in our case, from the datasets representation.

Looking at a topic map an ontology designer can think of it as having two distinct parts: ontology and an object catalog (instances). The ontology is defined by topic types, association types, occurrence types, role types, etc.

The catalog is composed of a set of pointers to information objects that are present in the resources and are linked to the ontology. So, a specification in XS4TM is composed of two parts:

- **Ontology**: the definition of the ontology requires in XS4TM the same effort as in XTM; it is necessary to specify every topic type, association type, occurrence type, ...;
- **Instances**: the instances definition describes each topic and association that will be extracted from the intermediate representation.

The XS4TM Context Free Grammar is based in XTM 1.0 [13]. The ontology and instances elements have the same syntax as the *topicMap* element in XTM model.

6 A processor that scans the input data sources to get desired data into the datasets, in agreement with an XSDS specification.

3.4.XS4TM Processor

This component uses the XS4TM specification and retrieves the information it needs to build the ontology from the datasets. It is an interpreter that takes advantage of the information organization in datasets (an internal universal representation for extracted data) and generates all the associations between the relevant topics according to XS4TM.

The XS4TM processor's behavior can be described in three steps: reads the XS4TM specification and extracts from the datasets the topics and associations found; creates the topic map; finally, stores it into a database or an XTM file.

4.XTCHE – A TOPIC MAPS CONSTRAINT LANGUAGE

When developing real topic maps, it is highly convenient to use a system to validate it; this is, to verify the correctness of the actual instance against the formal specification of that family of topic maps (according to the intention of its creator).

So, a specification language that allows us to define the schema and constraints of a family of Topic Maps is necessary. A list of requirements for the new language was recently established by the ISO Working Group – the ISO JTC1 SC34 Project for a Topic Map Constraint Language (TMCL) [11]. XTche language meets all the requirements in that list.

XTche [9] is designed to allow users to constrain any aspect of the topic map; for instance: topic names and scopes; association members, roles and players allowed in an association, instances of a topic (enumeration), association in which topics must participate, occurrences cardinality, etc.

Like XTM, XTche specifications can be too verbose; that way it is necessary to define constraints in a graphical way with the support of a visual tool. To overcome this problem, XTche syntax follows the XML Schema syntax; so, any XTche constraint specification can be written in a diagrammatic style with a common XML Schema editor. At the end the textual output of that edition (XML Schema code) should be processed to obtain a TM-Validator.

4.1.XTche Processor and TM-Validator

A XTche specification, listing all the conditions (involving topics and associations) that must be checked, specifies the Topic Map validation process (TM-Validator), enabling the systematic codification (in XSL) of this verification task. In those circumstances we understood that it was possible to generate automatically the validator developing another XSL processor to translate an XTche specification into the TM-Validator XSL code.

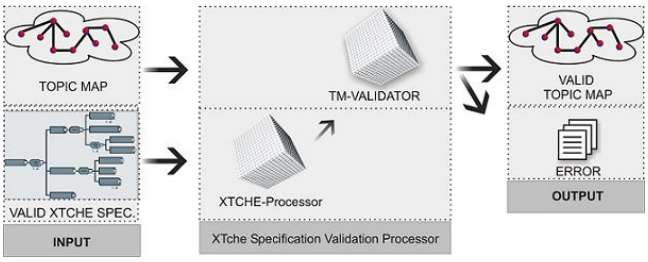

Figure 4: XTche Validation Process

According to Figure 4, the XTche processor is the TM-Validator generator; it takes a topic map constraint specification (an XML Schema, written according to the XTche language), and generates an XSL stylesheet (the TM-Validator) that will process an input topic map in order to verify its correctness.

5.ULISSES

Ulisses can be seen as a website generator from a XTM document (the "source" topic map) – this explains why we decided to integrate it as the last layer of Metamorphosis. It was conceived to be a autonomous (it can be used outside of Metamorphosis context) and simple way of creating full sites, with design, content and topical links; however, the layout of the site generated can be customized (page design, colors, . . .) to satisfy the specific user needs. Allowing the navigation on a conceptual network (an ontology described by the source topic map), Ulisses can be seen as a useful tool to develop the so called semantic web.

The basic idea behind the website generation is to create one HTML page for each topic or association. Hyperlinks are then used to connect related topics or topics and associations. A navigation menu, allowing to go back to the home page or to choose another view of the topic map, is always present in every page.

As told above, each topic or association name displayed in one HTML page is a hyperlink to the respective page, thus implementing the conceptual navigation over the semantic network described by the topic map. Ulisses' working is shown in Figure 5.

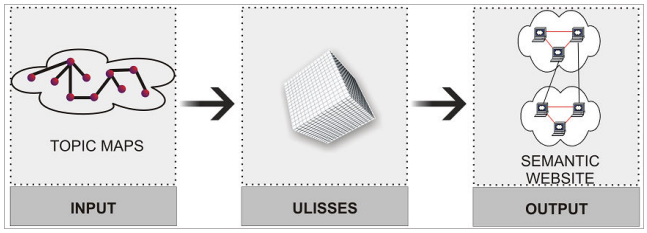

Figure 5: Ulisses Architecture

We developed three different versions of Ulisses: Ulisses I and II read the input from a XTM text file, while Ulisses III takes as input an OntologyDB (see above, sec. 3).

Concerning the generation strategy, the original version (Ulisses I) is a static generator—it processes just once the XTM file and creates at that time all the website pages; the generation is time consuming and the site directory huge, however the topic map navigation is very fast.

The drawback of that approach is that any change on the "source" TM implies the complete regeneration; otherwise the navigator becomes inconsistent/obsolete. To overcome that problem, the other two versions follow the opposite approach, implementing a dynamic generation; the first page (the homepage) is created at generation time and the others are created by need at navigation time.

6.RELATEDWORK

In terms of related work [16] we did not found an environment that can be compared to Metamorphosis. So, the comparisons below are among the main Metamorphosis' modules and their related work.

TSIMMIS [14] is a project that aims to provide tools for accessing, in an integrated fashion, multiple information sources, and to ensure that the information obtained is consistent. TSIMMIS gives a centralized view of the information that is distributed in the information system. Oveia was developed to allow a conceptual navigation over the heterogeneous information systems. This conceptual navigation is driven by an ontology specified from metadata extracted from information systems.

In another comparison, KAON REVERSE [15] has advantages concerning the use of a graphical interface for the specification of the ontology against Oveia. It also allows the use of reverse engineering of data sources to help creating the mapping. On the other side, Oveia is more flexible concerning data source formats and the specification process. To represent the ontology, KAON REVERSE adopts RDF; Oveia generates ontologies and stores them in an ontology database (*OntologyDB*) or in an XTM file.

AsTMa! [1] is another Topic Maps constraint language that has a mechanism to validate a topic map document against a given set of rules, like XTche language. That language has logic operators like NOT, AND and OR, simple logical quantifiers, and regular expressions.

When a comparison between XTche and the related works is done, some advantages is detected: XTche has a XML Schema-based language, a well-known format. In addition, XTche allows the use of an XML Schema graphical editor, like XMLSpy. With the diagrammatic view, it is easy to check visually the correctness of the specification.

7.CONCLUSION

Nowadays, data handled by an institution or company is spread out by more than one database and lots of documents of different types. To extract the information implicit in that data, it is necessary to pick parts from those various archives. To obtain a general overview, those information slices should be gathering. Different approaches can be followed to achieve that integration, ranging from the merge of resources till the fusion of the extracted parts. In this paper, we introduce Metamorphosis – a Topic Maps oriented environment to generate conceptual navigators for heterogeneous information systems – and we argue that Metamorphosis can be used to achieve the referred interoperability.

Metamorphosis let us achieve the semantic interoperability among heterogeneous information systems because the relevant data, according to the desired information specified through an ontology, is extracted and stored in a topic map. The environment validates it against a set of rules defined in a constraint language.

That topic map provides information fragments (the data itself) linked by specific relations to concepts at different levels of abstraction. Note that not all data items need to be extracted from the sources to the Topic Map. We only extract the necessary metadata to build the intended ontology. This ontology will have links to enable a browser to access all data items.

Thus the navigation over the topic map is led by a semantic network and provides a homogeneous view over the resources – this justifies our decision of call it semantic interoperability [10].

Although developed for use in our main working area – XML documents processing applied to Public Archives and Virtual Museums – we are convinced that Metamorphosis can be applied with similar success in the general area of information system for data integration, analysis, and knowledge exploitation.

8.ADITIONAL AUTHORS

Jonas Bulegon Gassen, UNIFRA - Centro Universitário Franciscano, Rua dos Andradas, 1614, Santa Maria, RS, Brazil, 97010-310. jbgassen@gmail.com

Rogério Corrêa Turchetti. UNIFRA - Centro Universitário Franciscano, Rua dos Andradas, 1614, Santa Maria, RS, Brazil, 97010-310. turchetti@unifra.br.

9.REFERENCES

[1] R. Barta. AsTMa! Bond University, TR., 2003. http://astma.it.bond.edu.au/constraining.xsp.

[2] M. Biezunsky, M. Bryan, and S. Newcomb. ISO/IEC 13250 - Topic Maps. ISO/IEC JTC 1/SC34, December 1999. http://www.y12.doe.gov/sgml/sc34/document/0129.pdf.

[3] R. Bourret. XML and Databases, 2005. http://www.rpbourret.com/xml/XMLAndDatabases.htm.

[4] E. Freese. Using Topic Maps for the representation, management and discovery of knowledge. In XML Europe 2000 Proceedings. http://www.gca.org/papers/xmleurope2000/papers/s22-01.html, June 2000.

[5] L. M. Garshol and G. Moore. Topic Maps – Data Model. In ISO/IEC JTC 1/SC34. http://www.isotopicmaps.org/sam/sam-model/, January 2005.

[6] N. Guarino. Formal Ontology and Information Systems. In Conference on Formal Ontology (FOIS98), 1998. http://www.ladseb.pd.cnr.it/infor/Ontology/Papers/FOIS98.pdf.

[7] N. A. Kipp. A mathematical formalism for the topic maps reference model. Draft paper submitted to ISO/IEC JTC1/SC34 Committee, 2003. http://www.isotopicmaps.org/tmrm/0441.htm.

[8] N. Leavitt. Whatever happened to object-oriented databases?, 2000. IEEE Computer.

[9] G. R. Librelotto, J. C. Ramalho, and P. R. Henriques. Constraining topic maps: a TMCL declarative implementation. In Extreme Markup Languages 2005: Proceedings. IDEAlliance.

[10] P. Mitra and G. Wiederhold. An algebra for semantic interoperability of information sources. In BIBE '01: Proceedings of the 2nd IEEE International Symposium on Bioinformatics and Bioengineering, page 174, Washington, DC, USA, 2001. IEEE Computer Society.

[11] M. Nishikawa and G. Moore. Requirements for a Topic Map Constraint Language JTC 1 NP Number. ISO/IEC 19756. ISO/IEC JTC 1/SC34 N0405, 2003. http://www.y12.doe.gov/sgml/sc34/document/0405.htm.

[12] J. Park and S. Hunting. XML Topic Maps: Creating and Using Topic Maps for the Web, volume ISBN 0-201-74960-2. Addison Wesley, 2003.

[13] S. Pepper and G. Moore. XML Topic Maps (XTM) 1.0. TopicMaps.Org Specification, August 2001. http://www.topicmaps.org/xtm/1.0/.

[14] M. Rys. TSIMMIS – The Stanford-IBM Manager of Multiple Information Sources, April 1998. http://www-db.stanford.edu/tsimmis/tsimmis.html.

[15] R. Volz. KAON REVERSE, 2003. http://kaon.semanticweb.org/alphaworld/reverse/view.

[16] H. Wache, T. Vögele, U. Visser, H. Stuckenschmidt, G. Schuster, H. Neumann, and S. Hübner. Ontology-based integration of information—a survey of existing approaches. In H. Stuckenschmidt, editor, IJCAI–01 Workshop: Ontologies and Information Sharing, pages 108–117, 2001.

[17] A. Wrightson. Topic Maps and Knowledge Representation. Ontopia, February 2001. http://www.ontopia.net/topicmaps/materials/kr-tm.html