Universal Indexing System for Structured Knowledge

Anabel Fraga, Juan Llorens and Karina Robles

Universidad Carlos III de Madrid, Departamento de Informática
Av. Universidad 30, Leganés, 28911, Madrid, Spain

Abstract. Knowledge is one of the main assets that humans have, the knowledge achieved in one area may be applied in another different area; all that you need is to remember it and adapt it to the new area or problem. If we apply this concept in computer science, it is amazing to realize that knowledge could be a powerful asset to store (remember) and reuse (adapt). Knowledge could be structured using different kind of Knowledge Organization Systems (KOS). If it is possible to index any kind of structured information in a repository that might support also a retrieval system, then it will be retrieved any knowledge improving the Reuse process and reducing costs at last.

1 Introduction

In previous research [13] the major points of failure of systematic reuse has been shown. A new perspective of reuse going back to the origins has been shown as well [13], it allows the process of improving retrieval techniques and methods, dropping investments costs, including traceability in the process and fully integrated into the software development process. Retrieval is one of the major "lost" activities in the reuse process. Diverse proposals arise in order to solve it, and diverse repositories and libraries are supporting storage and retrieval. Domain Analysis is another example trying to solve the retrieval issue, but in this case cost is really high because everything must be modeled a priori for further retrieval. It is a problem in the industry because it is almost forgot, reuse is not applied because of costs and ROI is low or negative in some cases.

As mentioned, proposals are present nowadays for retrieval and storage functions, but indexing is still on top. An indexer for each type of information is needed, so each type could be retrieved. And here a problem is foreseen: for each type of information an indexer must be developed. If an indexer for any kind of information could be developed then the process of reuse could be really improved reducing cost at last.

Following the research line, the main focus of it is the study of techniques, rules and development of a universal indexer, a transformer autonomous of the kind of information.

Universal Reuse [13] is the notion of knowledge reuse independently of the kind of information, the user that demands the need or even the context where it must be reused. Knowledge is an asset important for everyone, particular or company, and it has a peculiar characteristic, it is an asset that is possible to reuse in different situa-

Fraga A., Llorens J. and Robles K..
Universal Indexing System for Structured Knowledge.
DOI: 10.5220/0003704900670074
In *Proceedings of the 2nd International Workshop on Software Knowledge* (SKY-2011), pages 67-74
ISBN: 978-989-8425-82-9

tions, by diverse people demanding the reuse. It requires a special treatment or process in order to be reused in any context with the organization.

The reminder of this paper is structured as follows: Section 2 explains the Universal Reuse Representation Schema, Section 3 explains the Universal Reuse Indexing schema, Section 4 explains an overall of the Universal reuse Retrieval system, Section 5 explains a validation of the Universal Reuse Indexing system as core activity of the general process, and finally Section 6 enlighten some conclusions.

2 Universal Reuse Representation

The difference between information and knowledge is not clearly marked yet. It can be usually considered that information refers to general data expressed by numbers, words, images, sounds and so on, while knowledge refers to learned information, even by humans or computers. Then strictly speaking, knowledge could be very abstract. Knowledge is very difficult to accumulate, be sought and be integrated for new needs. One of the basic problems with different types of knowledge is that reusers do not always get what they need from repositories, for reasons that have to do in part with how repositories are created, in part with not up-to-date retrieval techniques, and with almost not existing solutions for smart merging and integrating knowledge within other knowledge. This is a big part of the "window" to be covered by the Knowledge Reuse area [3][4][5][6][7]. A well modelled knowledge implies a well retrieved knowledge later, and the time spend classifying knowledge will imply less time in the retrieval process. So, modelling is a challenge because it must be universal in the assumption of a universal reuse.

The Knowledge Organization Systems (KOS) domain is a formal and well studied area aiming with creating accurate structured knowledge, where Ontologies have a fundamental role. Ontologies play a great role in Knowledge Reuse as they could serve as repositories or even be reusable assets as well [8] [9] [10] [11].

For us, the structured information can be defined as information that have a data model (metamodel) that explains unambiguously (explicitly) and entirely (completely) the contents that its creator intended to. For us it could be called also Computable Structured Information due to the clarification of the passage from natural science to computer science [14]. It means the metamodel and its content must be represented in the same schema in order to be linked and retrieved later on.

An universal schema is important in the success of the reuse process, this schema must be able to keep the information and its metamodel representation, if both levels of representing information could be kept in the same schema, then we could call it as an Universal Schema. The problem to be solved in the case of modelling is that we need to model any kind of information. For that, we must have a generic metamodel information. Diverse kind of schemas for representing information are available, but RSHP [1] [2] is available schema for this research and we will use it in order to represent any kind of information.

RSHP stands for "RelationSHiP" [12], and it was designed to jointly represent all different types of information using a common meta-model that allows all possible

information models to be stored in the same repository, in order to enable traceability between the represented information artifacts.

The philosophy of RSHP is based on the ground idea that knowledge is in essence, related facts, and therefore, it is necessary to bring the relationship itself to the highest priority of a representation model. As a result of this premise, "In order to represent information in the RSHP representation model, the main description element to be found within the container of the information to be represented should be the relation-ship. This relationship is in charge of linking concepts" [1] [2].

The big deal here is that RSHP could help us to represent the information without lost, using its metamodel, because of that it is possible that some stages in the reuse process could be solved thanks to the use of RSHP. If we can solve the representation in RSHP of any kind of information some process like retrieval are solved by reciprocal, because RSHP has a retrieval process solved for the repository and information that it manages. And in the case that each stage of the general reuse process could be solved in the side of the RSHP schema, then it will be solved for the process itself. Thanks to this schema, knowledge could be stored in a universal repository.

3 Universal Reuse Indexing

The Universal Indexing method at design level must commit the analysis made and rules established to this closing stages.

If we think of indexing nowadays, each kind of information has its own indexer, the indexing process is attached to its information kind, and it means an indexer is needed for each kind of information – i.e. for each metamodel. If the indexing activity is limited to structured information, but not free text which is managed by Natural Language Processing techniques (NLP); then it is possible to design an algorithm for the accomplishment of this task.

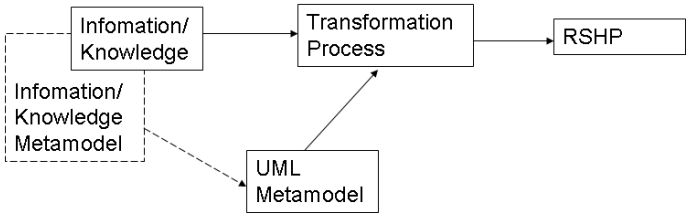

Fig. 1. Indexing process flow diagram.

The indexing process could be defined as a process of transforming knowledge into a storage model without lost of information. So far, not losing information is a very difficult task. First, some rules to transform information must be implemented and these rules must consider a storage model able to keep any kind of knowledge.

As proposed, RSHP is a well suited storage schema and it helps in the process of preserving information due to its inherent properties as a generic representation model. One of the main problems to deal with is to index or transform the knowledge

without lost of information. For that, the rules must be a complete set of transformation rules.

As shown in Figure 1, the information to be indexed needs the existence of its metamodel, in order to keep the meta-relationships between the concepts thanks to the transformation process that follows rules, it transforms the information and its metamodel into the universal repository schema.

The metamodel could be represented in UML (Unified Modeling Language) or it could be extracted if not available using the XML (eXtensible Markup Language) files structure, when talking of structured information.

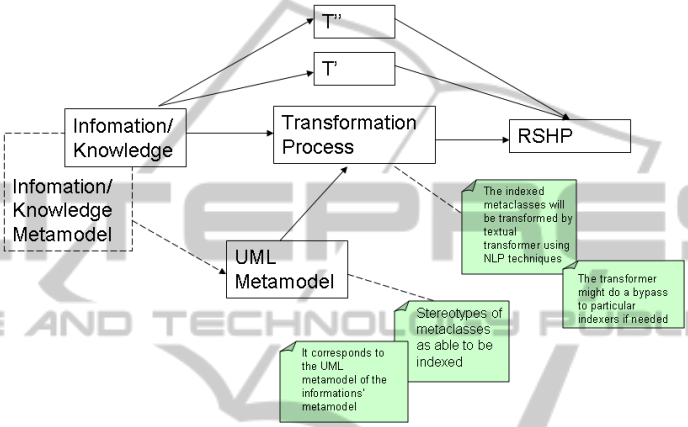

Fig. 2. Enhanced Indexing process flow soft diagram.

Even more, the indexing process could be expanded using auxiliary indexers, as for example NLP indexers (see Figure 2). As shown in Figure 2, any type of information, as: UML, XML, structured text, code, and so on; will have a treatment of a kind of information *Ii* in a generic form. Each *Ii* must be stored and later retrieved for adapting it to a new context.

As shown in Figure 3, the universal indexing must transform any kind of information (*Ii*), without lost of information. For that, a set of rules has been proposed, these rules are implemented in the universal transformer, and it is called *Tu*. Each *Ii* requires a metamodel for later transformation, without a metamodel it won't be possible to obtain the relationships between information. Thanks to the metamodel, information relationships are extracted and semantic knowledge is kept in the repository.

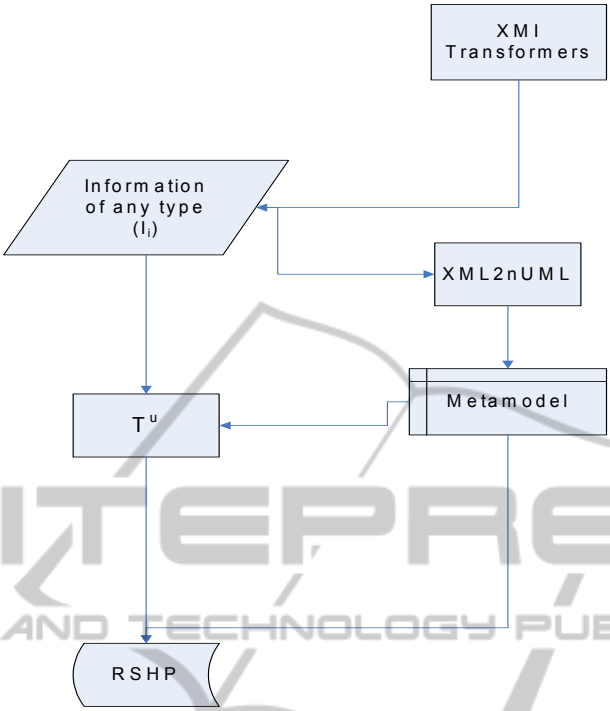

Fig. 3. Universal Indexer flow chart.

Universal indexing must transform any kind of information *Ii*, without lost of information. For that, a set of rules is proposed, these rues will be implemented in the universal transformer or indexer, and it will be called *Tu*. Each *Ii* requires a metamodel for later transformation, without a metamodel won't be possible to obtain the relationships between information. Thanks to the metamodel, information relationships are extracted and semantic knowledge is kept in the repository.

Each *Ii* is a structured information representation, so it could be kept in XML widely used for structured information interchange. The metamodel for each *Ii* will be represented using UML as modeling language. UML could be represented in a structured schema called: XMI (XML Metadata Interchange). And it will be used in the universal transformer as input.

In some cases, the metamodel is not available, so a reverse engineering process is needed. It is called XML2UML indexer. The metamodel is extracted using XML information as foundation. The metamodel could be described by a human, but in this case the process has been designed in order to be independent for better validation. If a human is involved, of course, the process will be better and improved. A tool for creating the metamodel is a good solution in this case. A pseudo code is shown as follows:

Algorithm: Indexer(Metamodel, Information_*source*)
--Pseudo Code--

```
1  for each Information_source corresponding to Metamodel:
     // Initialization
If not( Existing_Metamodel)
// Parse Metamodel
   Nodes := Parse_Metamodel(nodes in  Information_source)
   Load_XML_structures_in memory(Nodes)
   GenMetamodel := Generate_Metamodel(Information_source)
Else if (Existing_Metamodel)
   GenMetamodel := Load_Metamodel(Metamodel)
endIf
If isNew(GenMetamodel)
   IdMetamodel := Save_Metamodel(GenMetamodel)
Else
  IdMetamodel := Retrieve_Metamodel_Id(GenMetamodel)
endIf
LoadedMetamodel := Load_metamodel _inMemory(IdMetamodel)
Parse_content(Information_source, LoadedMetamodel)
// Parse  Instances
```


4 Universal Reuse Retrieval

The main problem is to be able to retrieve any kind of information in this stage, and it could be solved in the RSHP side. As explained, the use of RSHP as a universal storage model aids in the retrieval stage. RSHP provides powerful retrieval capabilities, after storing knowledge it could be retrieved in a simple way: as a graph based on relationships and concepts. It is the basic idea of knowledge. Knowledge is based on concepts and relationships between them, the more relations you have then the most you retrieve. Retrieval then relies on previous steps. Retrieval algorithms are important, new approaches and methods for retrieve in a fast and in a best semantic way with better rates of Precision/Recall must be done, but the most important fact is that knowledge must be stored and indexed in an appropriate manner in order to retrieve it with fulfilment.

5 Validation

Some experiments were thoughts for validating these ideas, it consists of four different experiments, each experiment tries to test the hypothesis and find conclusions and improvements for the Universal Indexer. The experiment intends to probe but also to improve the solution given in this research work.

First, one kind of information and its metamodel will be indexed using four different methods. For each of the methods, metrics will be taken: Economic metrics for

developing and using the method, and time processing the information. The experiment will be done in three stages, each stage will increment the metamodel including a new kind of relationship, and after each stage the metrics will be evaluated. It means each indexer must change in order to adapt to the new requirement.

Second, one kind of information and two metamodels will be indexed using four different methods. For each of the methods metrics will be taken, as will be done for the first experiment. At this time optimization of previous indexers is needed also.

Third, now search is on focus, after indexing retrieval must be measured as well. Five different search patterns, some including semantics, will be defined and applied on the indexed information for each of the previous steps. Metrics will be taken: extracted information, precision, recall, E, F, ASL and time for query resolution.

Fourth, given a set of XML files downloaded using one well known search engines, a set of two queries will be settled and the search will be performed using the Universal Indexer at local and the Search engine in Internet. The result of this experiment consists of comparing data extracted for both queries, one with semantic of the metamodel of the indexer files.

6 Conclusions

The classical systematic reuse process failed in the industry environment because of the huge investment needed to be accomplished by practitioners. Low or negative ROI ratios became one of the key problems for its wide-spreading. Aside ad-hoc reusers also gained a certain level of success but the accomplishment level is low, reuse is only applied to code, dlls and components, and the practice of this reuse has been chaotic.

Industry would get worth of dealing with any kind of knowledge, in any context, and by any user: anything, anywhere, and anybody. For that reason, we offer the concept of Universal Indexer and in previous research a Universal Reuse System as an open door to get all the benefits of theoretical reuse avoiding the well known drawbacks of systematic and ad-hoc reuse.

The whole process for reusing any kind of knowledge has to deal with: a universal representation model, a universal indexer, a universal retrieval and adaptation activities, a universal accessing, knowledge visualization and a universal reuse metric. All of these activities have to face the issue that each one could be applied to any kind of knowledge, in any context, and each activity might be required by any user.

This is a positioning paper, the whole system is under development, but future publications will offer this concept to industry and its experiments' results.

References

1. J. Llorens, J. Morato, G. Genova. RSHP: An information representation model based on relationships. In: Ernesto Damiani, Lakhmi C. Jain, Mauro Madravio (Eds.), Soft Computing in Software Engineering (Studies in Fuzziness and Soft Computing Series, Vol. 159), Springer 2004, pp 221-253. Available for reviewers in ftp://

www.ie.inf.uc3m.es/llorens/ICSR.zip
2. Llorens, Juan; Fuentes, José M.; Prieto-Diaz, Rubén; Astudillo, Hernán. Incremental Software Reuse. International Conference of Software Reuse (ICSR2006). Torino, Italy. 2006.
3. Knowledge acquisition and retrieval apparatus and method. http://www.patentstorm.us/patents/6611841-description.html US Patent Issued on August 26, 2003. [Last visited on 15th of November of 2008]
4. R. Davis, H. Shrobe, and P. Szolovits. What is a Knowledge Representation? AI Magazine, 14(1):17-33, 1993.
5. Arthur B. Markman: Knowledge Representation Lawrence Erlbaum Associates, 1998.
6. Ronald J. Brachman; What IS-A is and isn't. An Analysis of Taxonomic Links in Semantic Networks; IEEE Computer, 16 (10); October 1983.
7. John F. Sowa: Knowledge Representation: Logical, Philosophical, and Computational Foundations. Brooks/Cole: New York, 2000.
8. KOS: Knowledge Organisation Systems. http://www.db.dk/bh/lifeboat_ko/concepts/knowledge_organization_systems.htm [Last visited on 15th of November of 2010]
9. Bechofer S., Goble C. Thesaurus construction through knowledge representation. Data & Knowledge Engineering, 37, 25-45. 2001.
10. Hill et al. 2002. Integration of Knowledge Organization Systems into Digital Library Architectures. ASIST SigCR - http://www.lub.lu.se/SEMKOS/docs/Hill_KOSpaper7-2-final.doc. [Last visited on 15th of March of 2009]
11. Janée G., Ikeda S., Hill L. ADL Thesaurus Protocol v1.0. 2002. http://www.alexandria.ucsb.edu/thesaurus/protocol
http://nkos.slis.kent.edu/2002workshop/janee.ppt [Last visited on 15th of March of 2009]
12. J. Llorens, J. Morato, G. Genova, “RSHP: An information representation model based on relationships.” In: Ernesto Damiani, Lakhmi C. Jain, Mauro Madravio (Eds.), Soft Computing in Software Engineering (Studies in Fuzziness and Soft Computing Series, Vol. 159), Springer 2004, pp 221-253. Available for reviewers in ftp://www.ie.inf.uc3m.es/llorens/ICSR.zip
13. Fraga, A., Llorens, J. “Universal Knowledge Reuse: anything, anywhere, and anybody”. KREUSE2008/ICSR2008. International Conference of Software Reuse. International Workshop on Knowledge Reuse. Proceedings ISBN: 978-84-691-3166-4. Beijing, China, 2008.
14. John F. Sowa: Knowledge Representation: Logical, Philosophical, and Computational Foundations. Brooks/Cole: New York, 2000.