Adaptive Integration of Information

Christophe Nicolle, Christophe CRUZ
Laboratory Le2i
Université de Bourgogne
B.P. 47870, 21078
Dijon CEDEX, France
{christophe.cruz, cnicolle}@u-bourgogne.fr

Keywords: XML, data integration, OWL, RDF, CDMF.

Abstract: This paper suggests a new approach to improve the integration of information. This approach is based on the use of semantic adaptive graphs. The adaptive feature of our proposal makes it possible to manage two specific aspects related to information integration: the adaptation of information according to the user's access rights and the lifecycle of the integrated information.

1 INTRODUCTION

In the field of information systems, integration consists in unifying heterogeneous information sources for a given user. Since the Seventies, the scientific community has tried to build up this unified view through the proposal of various models, languages and architectures independently of an implementation (Navathe, 1982)(Chen, 1976). Today, the integration of information becomes a set of calls made to Web services, which are integrated into an XML document (Abiteboul, 2008). The resulting graph is built dynamically according to the results of the distant calls. Nevertheless, for each proposal, semantic heterogeneity remains an unsolved subjacent problem. In this field, the interest is to model a tacit knowledge which is related to the information. The suggestions evolve from the models to the metamodels, then towards the mediators and finally to ontologies (Liu, 2007). During this period, the structure is strongly influenced by the emergence of the object oriented models. Then, derived from the concept of graphs (Sowa, 1984), the concept of the semantic graph appears, as well as the concept of ontology (Guarino, 1994). In our example, the integration of information using ontology makes it possible to assemble two pieces of a puzzle according to the image which is printed on them without worrying about the shape of the pieces.

From the combination of XML, ontology and Web services, new proposals appear. For the structural part, the comparison of XML grammars becomes an important field of study. These proposals are the descendants of those carried out by Miller developed in 1993 (Miller, 1993). For the part that is concerned with information access, orchestration and choreography languages are developed to combine the Web services. For the semantical part, OWL and RDF languages become the angular stones of research on ontologies. The combinations of ontologies and XML are effective when they are used to integrate information with the objective of building a global system within the meaning of the first distributed architectures (Bell, 1992), i.e. the source systems will disappear. Only the built target system will be used. The formats of the source models are replaced by XML schema. The heterogeneous semantics of distributed information will be homogenized in a common thesaurus defined by ontology.

In the industrial field, the lifecycle of information is of fundamental importance in the integration process. This process is conditioned by the nature of the information but also by the way in which it is used. The nature of information can change according to the context of use. In this approach, integration is not finality but it is a continuous process. During this process it is possible to dynamically adapt the integrated information according to the lifecycle of local information from data source systems. To reach these requirements, this paper presents a new approach based on adaptive semantic graphs. These graphs model information as well as the contexts of use of this information and the lifecycle of this information for each context. Our model is based on a combination of various operators, such as RDF, OWL, SWRL and Named Graph. The nodes described in these graphs can represent multi-media information, operators on the graphs and sub-graphs.

The following section is made up of three parts. The first part gives an overview of the framework CDMF. The second part presents a brief description of the operators used to model the structural part of the graph. The third part presents the operators used to model the contextual part and the lifecycle of information. The last part concludes this paper.

2 CDMF OVERVIEW

The CDMF framework gives a set of operators and modeling structures which allow to deal with temporal and contextual requirements of the adaptive integration of information. The applicative environment developed for CDMF is composed of several parts: the engine, the configuration graph called "SpaceSystem" and the interface (called API CDMF).

- The engine is composed of the Data Modeling Layer and the Context Modeling Layer. The first layer relates to the inference engine which is used to infer and to check the data modeled by the CDMF modeling operators. It is made up of the implementation of the graph combination operators (such as AddGraph, RemoveGraph, MapGraph, etc.) and of the element SystemGraph. The second layer is used to manage the contexts in the CDMF architecture.
- The SpaceSystem constitutes a configuration space used by the inference engine. This space contains a set of graphs SystemGraph containing the initialization data.
- The API CDMF proposes a set of functions giving access to the various functionalities of the CDMF applicative environment.

Due to lack of space, we restrict our presentation to the Data Modeling Layer (DML) and the Context Modeling Layer (CML). DML defines a reduced set of operators allowing the semantic modeling of information. These operators are based on RDF specifications. CML is dedicated to context modeling and graph manipulation.

3 THE DATA MODELING LAYER

DML is a language composed of a set of operators derived from RDF, OWL and SWRL. DMF makes it possible to describe classes and properties. These classes and these properties can then be used in statements (formulas) using operators of implication, intersection, union, etc. The operator of implication allows to constitute rules which express constraints on these sets of individuals. The operators composing this layer are

- `dmf:Class` defines a class.
- `dmf:Property` defines a property.
- `dmf:Equal` defines the equality of two resources.
- `dmf:Var` defines variables used in the logical formulas.
- `dmf:Predu` defines unary predicates.
- `dmf:Predb` defines binary predicates.
- `dmf:Equiv` defines two predicates as equivalent.
- `dmf:And` defines the intersection.
- `dmf:Not` defines the negation.
- `dmf:Or` defines the union.
- `dmf:OrX` defines the exclusive disjunction.

For the following statements, the operator of implication `dmf:Imp` is used to represent various operators used in OWL. `dmf:Imp` is equivalent to the operator `ruleml:Imp` defined in SWRL. This SWRL operator is derived from the RuleML formalism.

- `Imp(p1(?x,?y),p2(?x,?y))`: defines p1 as a sub-property of p2.
- `Imp(p(?x,?y),And(A(?x),B(?y))`: defines restrictions of the type for the subject and the object of a property. Here, all the subject instances of the property p are of type A and all the object instances of the property p are of type B
- `Imp(And(p(?x,?y),p(?y,?z)),p(?x,?z))`: defines a transitive property
- `Imp(p(?x,?y),p(?y,?x))`: defines a symmetrical property.
- `Imp(And(p(?x,?y),p(?x,?z)),Equal(?y,?z))`: defines a functional property. This feature indicates a single property. It is a short cut to declare a minimal cardinality being 0 and a maximal cardinality being 1.
- `Imp(p1(?x,?y),p2(?y,?x))`: defines p2 like the inverse property of p1.
- `Imp(A(?x),B(?x))`: defines A as a subclass of B.

- `Imp(And(p(?x,?y),p(?z,?y)),Equal(?x,?z))`: defines an inverse functional property.
- `Imp(And(A(?x),p(?x,?y)),B(?y))`: defines all the values of p as being of type B.
- `Imp(A(?x),[i,j]p(?x,?y))`: defines cardinalities which are defined by a couple of values between square brackets. The first value defines the minimal cardinality and the second value defines the maximum cardinality. In this statement, any element of the type A contains properties from i to j.
- `Imp(A(?x),p(?x,value))`: defines a default value for a class and its property. The elements of the type A have a property p whose value is "value".

The following example illustrates in an RDF/XML format the use of the DML operators for the construction of a schema. The two first lines create a class `Building` and a class `Room`. The third line creates a property `contains`. This example composes the `cdmf:model` which is part of the `dmf:systemgraph` presented in the next section.

```
1 <dmf:Class      rdf:ID='Building'/>
2 <dmf:Class      rdf:ID='Room'/>
3 <dmf:Property   rdf:ID='contains'/>
```

The next example presents a DMF graph modeling data according to the previous schema. Due to lack of space, the format is in RDF/N3. This example composes the `cdmf:graph` which is part of the `dmf:systemgraph` presented in the next section.

```
1 :G1 {
2  :room_1 rdf:type :Room
3  :building_1 rdf:type :Building
4   :building_1 :contains :room_1 }
```

The first line defines a data graph called G1. Line 2 defines two instances of class `Room` called `room_1`. Line 3 defines an instance `building_1` from the class `Building`. This instance is linked with `room_1` using the property `contains` in line 4. This example presents the static part of our approach. This part makes it possible to integrate heterogeneous information without taking into account the lifecycle of information.

4 CML

The Context Modeliing Layer (CML) has been developed to enhance the static part defined in the DML. CML is articulated in two parts. The first part defines the context of use for each DML graph. The second part defines a set of operators used to combine graphs and to limit data redundancy. CDMF organizes a set of properties into a graph. These properties are used to model control access (read/write) and to represent context. This element is called `cdmf:SystemGraph`. It is composed of properties that describe with the help of graphs the user's access conditions, the data model in which the data graph is structured and the data graph:

- `cdmf:graph` connects graph and data. These data are described according to the data model (which can be a combination between other graphs using CDMF operators).
- `cdmf:of` represents the context. This property defines a list of resources representing the access context.
- `cdmf:model` defines for a system graph the data model which is used. This data model defines elements which will appear in the graph.
- `cdmf:action` defines user's rights to access the data. (read/write/remove)

We derive from Named Graph the representation of contexts. This property makes it possible to define an RDF graph like a resource related to a context. The sub-graph modeling the context is defined by the property `cdmf:of`. In our proposal we used the context to construct an integrated sub-system according to the type of user. It enables us to build an integrated information management according to user environments. For example, it is possible to link the data graph G1 with a context defined by a user (Line 1) and a date (Line 2).

```
1 :G1 :auteur  'Christophe Cruz'.
2 :G1 :date    '08/23/08'.
```

The second part of CML is made of operators. The use of these operators allows the simplification of the management of the evolution of integrated information. Rather than to store a new version of information, we integrate the process of evolution of the information into a graph of operators.

- `cdmf:AddGraph` allows the union of two or several graphs. It has a property `cdmf:args`. This property is a list of RDF elements (`rdf:Bag`) which are graphs.

  To illustrate this operator, let us consider the first example G1 with a Building_1 which contains room_1. If the next day '08/24/08', the same user 'Christophe Cruz' updates this structure by adding a new room to the building, the following new `cdmf:SystemGraph G2` will be built. The `cdmf:graph` part will contain:

```
1 :G2   rdf:type    cdmf:AddGraph
2 :G2   cdmf:args   :li1
```

```
:li1  rdf:li     :G1
:li1  rdf:li     :G1b
:G1b {
      :room_2 rdf:type :Room.
      :building_1 :contains :room_2 }
```

The `cdmf:of` part contains the reference to the context:

```
:G2 :auteur    'Christophe Cruz'.
:G2 :date      '08/24/08'.
```

The updates of the integrated information are stored in the graph. In this example, it is possible to go back to a former version of the integrated system.

- `cdmf:InterGraph` carries out the intersection on the sets of triplets of each graph. It has two properties `cdmf:arg1` and `cdmf:arg2`, which represent the two graphs on which the intersection must be calculated
- `cdmf:CompInterGraph` makes it possible to determine which part of the triplet is concerned by the calculation of the intersection.
- `cdmf:MapGraph` defines a graph of correspondence. This graph is a transformation of a graph into another graph using rules of correspondences. It has two properties `cdmf:src` and `cdmf:map` indicating the graph source and the graph where the rules of transformation are defined.
- `cdmf:RemoveGraph` makes it possible to remove a part of a graph. It has two properties `cdmf:src` and `cdmf:rem`. The second property constitutes the set of the triplets to be withdrawn from the graph indicated by the first argument.

We tested several combinations of operators in order to:

- preserve the history of the integrated information updates
- adapt the model of integrated data according to the new source models which are added during the lifecycle of the system.
- ensure the migration of the data according to the evolution of the integrated data model.
- build interfaces adapted to the profile of the user.

An interface is a sub-graph which connects the data, the process and a graphic charter according to the rights and the context of the user.

4 CONCLUSION

This paper made an attempt at presenting a state-of-the-art related to the integration of information in the field of information systems. The authors tried to underline the difference between the academic proposals and the actual industrial realities. The paper showed that integration is not a final process. The integration of information in the industrial world requires a perpetual update of the integrated system while taking into account the lifecycle of information and the user context. To meet these needs, this paper presented an integration method based on adaptive semantic graphs. These graphs make it possible to facilitate the process of integration while proposing mechanisms to update the integrated information and its context representation.

Our proposal was implemented into a Web collaborative platform dedicated to facility management. In this field, the lifecycle of the building is made up of four steps: design, construction, exploitation and maintenance. In this platform, the local users preserve the use of their local systems. Each local user can obtain an integrated sight of the building in the form of a 3D mock-up generated from the operators of CDMF. Currently, more than 6 million m^2 of building surface are being integrated.

REFERENCES

Abiteboul, S., Benjelloun, O., Milo, T., *The Active XML project: an overview*, VLDB Journal, 2008.

Bell, D., Grimson, J., *Distributed Database Systems*, Int. Computer Science Series, Addison-Wesley Publishing Company, 1992.

Chen P.P., *The Entity-Relationship Model Toward a Unified View of Data*, ACM Transaction on Database Systems, Vol.1, N°1, 1976.

Guarino, N., Carrara, M., Giaretta, P., *An Ontology of Meta-Level Categories*. Proc. of the 4th Int. Conf. on Principles of Knowledge Representation and Reasoning, pp 270-280 Bonn, Germany, 1994.

Miller, R.J., Ionnadis, T.E, Ramkrishnan, R., *The Use of Information Capacity in Heterogeneous Systems: Bridging Theory and Practice*, Proc. of the 19th VLDB Conf., Dublin, Ireland, 1993

Liu, Q., Huang, T., Liu S., Zhong, H., *An Ontology-Based Approach for Semantic Conflict Resolution in Database Integration*. Journal of Computer Science. Technology Vol. 22 N°2, pp 218-227, 2007.

Navathe, S.B., Gadgil, S.G., A, *Methodology for View Integration in Logical Database Design*, Proc. of the 8th Inter. Conf. on VLDB, pp142-164, 1982.

Sowa, J.F., *Conceptual Structures: Information Processing in Mind and Machine*, Addison-Wesley 1984.