

# Ontological Engineering in Data Warehousing

Longbing Cao, Jiarui Ni, and Dan Luo

Faculty of Information Technology, University of Technology Sydney, Australia  
{lbcao, jiarui, dluo}@it.uts.edu.au

**Abstract.** In our previous work, we proposed the ontology-based integration of data warehousing to make existing data warehouse system more user-friendly, adaptive and automatic. This paper further outlines a high-level picture of the ontological engineering in data warehousing. Its basic theory includes building ontology profiles for warehousing in terms of domain ontology and problem-solving ontology, analyzing ontological commitment and semantic relationships of ontological items, aggregating, transforming, mapping, querying and discovering semantic relationships and/or ontological items crossing multiple relevant ontological domains. We introduce and illustrate them very briefly in terms of a web-based electronic institution F-Trade.

## 1 Introduction

The existing data warehousing is based on a two-step process. First, data dispersing in relevant enterprise information systems (EIS) is extracted, transformed and loaded into a data warehouse (DW) server according to predefined algorithms and data models. Then analytical engines for predefined, ad hoc, OLAP or data mining analyses are linked to the DW to present reporting results based on the existing models. This process makes the current DW inflexible, non-ease of use and heavily human-dependent. In our previous work [1-6], we have addresses a series of issues and proposed ontology-based approaches to more flexible integration of EIS, more adaptive data modeling and easier personalization in a business-oriented way.

The basic theory of ontological engineering of data warehousing is as follows. We first build the ontology profile for the DW in terms of a specific problem domain. Second, we analyze the ontological commitment and semantic relationships among ontological items. Third, certain specifications are developed to depict and represent the ontology profile. Fourth, a proper architecture must be built to integrate the multiple domains in warehousing. Finally, semantic rules are developed to aggregate, transform, map and query the multi-domain ontological items in ontological DW.

## 2 Ontology Profile in Warehousing

A fundamental step in the ontology-based warehousing is to build a comprehensive ontology profile which transforms a domain problem and its problem-solving (DW)

system to domain ontology (DO) and problem-solving ontology (PSO). Figure 1 shows the structure of such a ontological profile, where the PSO is further divided into categories as task, method, business logic (BL) and resource ontology.

As the first step, *domain ontology* (DO) extracts the essence of conceptual items in a problem domain. DO consists of reusable vocabularies of concepts and relationships within a domain, activities taking place in that domain, and theories and elementary principles governing that domain. Taking the capital markets as an instance, Figure 2 illustrates an excerpt of the relevant concepts and their relations in stock market microstructure. In the stock, the Financial Order is categorized as Limit Order, Market Order or Stop Order; while for every trade, it consists of attributes such as Price, Dealer, Date, Time, Volume and so forth.

Due to the complexity of data warehousing and customer feedbacks in the real world, we emphasize the involvement of business requirements and domain knowledge. Therefore, we highlight relevant concepts and their relationships associated with solving the domain problem in terms of business-oriented *task, method, business logic* and *resource ontologies*.

Taking the financial trading and mining analysis system F-Trade [7] as an instance, Figure 3 illustrates the main high-level tasks the system must conduct to support the designed goals of the system. The whole system is decomposed into five main tasks (corresponding to five Centers): system administration, algorithm plug-in and evaluation, service providing, system control and evolution, and user interaction and management. Every task may be divided into subtasks. Correspondingly, ontological items will be specified for them respectively.

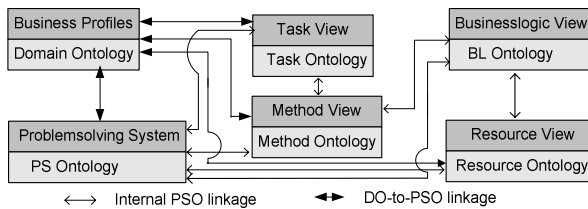


Fig. 1. Ontology profile in the warehousing

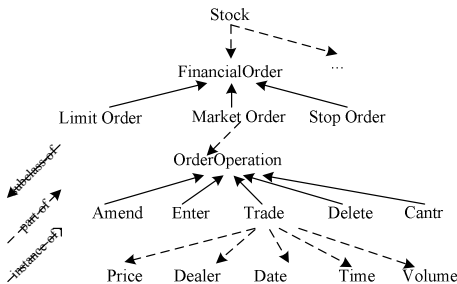


Fig. 2. Domain ontology in capital market

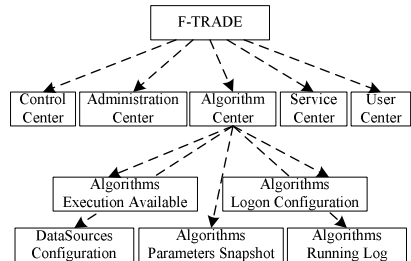


Fig. 3. Task ontology

A set of methods may be called to fulfill a specific task. *Method ontology* describes ontological attributes related to a method. For instance, the task Algorithms Logon Configuration can be handled via man-machine interaction or interface agents. Every method is relevant to one or many BL. A BL is a functional and computational unit in a problem-solving system. It may be divided into smaller units on demand. It involves system architecture and design patterns, and working flow and the processes. *BL ontology* consists of ontological items used for the management and execution of the above aspects in the problem-solving system.

For instance, in the F-Trade, each BL unit is decomposed into four classes as action, model, controller and view in terms of MVC design patterns, which are interacted via business logic relationships. For instance, the activity of executing an algorithm is decomposed into four classes as *AlgoExeAction*, *AlgoExeModel*, *AlgoExeController* and *AlgoExeView* (shown in Figure 4) linked via some business logic interactions.

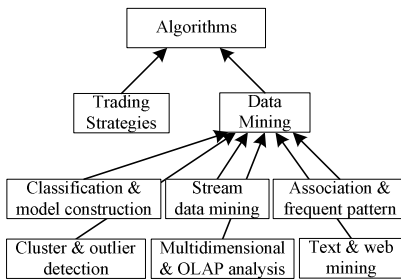


Fig. 4. Business logic ontology

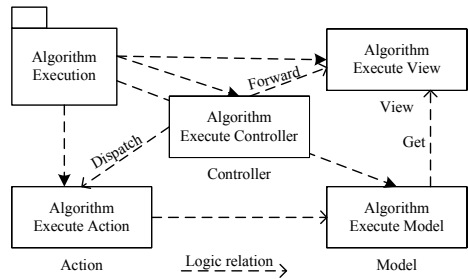


Fig. 5. Resource ontology

Resources in a DW system include: (i) domain database storing business data, (ii) knowledge base storing all algorithms, data models and rules, and (iii) system base keeping all configuration and management information of the system, and so forth. *Resource ontology* captures all objects and relations in all these resources.

Here, we illustrate some resource ontology related to the task of algorithm management in the knowledge base of the F-Trade. As shown in Figure 5, algorithms consist of trading strategies and data mining algorithms. With regard to data mining, algorithms can be categorized as classification & model construction, stream data mining, association & frequent pattern, cluster & outlier detection, multidimensional & OLAP analysis, and text & web mining according to some classification criteria. Following this framework, we can go deep into lower levels to define ontological items if applicable.

### 3 Ontological Commitment and Semantic Relationships

In building the mapping from a natural concept to its ontological item or between two ontological domains, an essential work is to deal with the synonymous and multivocal phenomena widely seen in business world. This is handled by *ontological commitment*. Ontological commitment defines an agreement to use a shared ontology

library in a coherent and consistent way. It permits ontologists to share and instantiate ontological items with committed freedom.

DEFINITION 1. An ontological commitment is a five-element tuple:

$$OC = (C, O, R, P, S).$$

Where,

$C = \{c_i \mid i \in I\}$  is a set of domain-specific concepts in a given domain, it could be some business key word/phrase specified by end users for their preferences.

$O = \{o_j \mid j \in J\}$  includes a set of candidate ontologies in domain or problem-solving ontological base;  $o_j$  is relevant or mapped to a given  $c_i$ .

$R = \{r_i \mid i \in I\}$  is a set of semantic relationships between  $c_i$  and  $o_j$  or items across domains.

$P = \{p_i \mid i \in I\}$  is a set of cardinality properties specifying how  $c_i$  is associated with  $o_j$ .

$S = \{s_i \mid i \in I\}$  optionally measures the similarity between  $c_i$  and  $o_j$ .

In the above definition,  $I$  and  $J$  are two separated positive integer sets. In addition,  $C$  and  $O$  can also refer to two varied ontological domains.

DEFINITION 2. An atom item of ontological commitment can be defined as follows.

$\langle \text{ontocommit} \rangle ::= (\langle \text{relation} \rangle (\langle \text{concept} \rangle, \langle \text{ontology} \rangle) : \langle \text{cardinality} \rangle [, \langle \text{similarity} \rangle])^*$

$\langle \text{relation} \rangle ::= (\text{Instantiation} \mid \text{Aggregation} \mid \text{Generalization} \mid \text{Substitution} \mid \text{Disjoin} \mid \text{Overlap} \mid \text{Association})$

$\langle \text{concept} \rangle ::= (\{ \langle \text{name} \rangle \mid \text{name} \in C \})$

$\langle \text{ontology} \rangle ::= (\{ \langle \text{name} \rangle \mid \text{name} \in O \})$

$\langle \text{cardinality} \rangle ::= (\text{SingleUnrestricted} \mid \text{SingleRestricted} \mid \text{MultipleUnrestricted} \mid \text{MultipleRestricted})$

$\langle \text{similarity} \rangle ::= (\{ s \mid s \in [0, 1] \})$

To manage the above mentioned mapping, we need to build corresponding mechanisms for the identification and aggregation of semantic relationships between ontological items across domains in the warehousing. Varying combinatory scenarios of natural terminologies and their notations furthermore result in different semantic relationships among ontological items. In [2,6], we identify seven types of semantic relationships from them. They are *Aggregation*, *Association*, *Disjointness*, *Generalization*, *Instantiation*, *Overlap*, and *Substitution*. For space limitations, please refer to [2,6] for details.

## 4 Ontological Representation

In Section 2, ontological items are visually presented as tree with terms and their relationships. On the other hand, formal or semi-formal specifications can present the ontological structure more precisely. In our work, we define ontological grammar in *description logics* (DL) since it supports intentional concepts and properties of concepts, and allows the construction of composite descriptions.

*/\*DL-based grammar for presenting ontology\*/*

$\langle \text{onto-item} \rangle ::= \langle \text{item-name} \rangle \mid$

$(\langle \text{logic-connective} \rangle \langle \text{onto-item} \rangle +) \mid$

$(\langle \text{bound-connective} \rangle \langle \text{bound-number} \rangle \langle \text{property} \rangle) \mid$

$(\langle \text{cardinality-connective} \rangle (\langle \text{property} \rangle \langle \text{constraint} \rangle)) \mid$

$(\langle \text{semantic-relationship} \rangle \langle \text{onto-item} \rangle + \mid \langle \text{property} \rangle +) (\langle \text{onto-item} \rangle + \mid \langle \text{property} \rangle +)$

$\langle \text{item-name} \rangle ::= \langle \text{symbol} \rangle$

<logic-connective> ::= (AND | OR)  
 <bound-connective> ::= (AT-MOST | AT-LEAST)  
 <cardinality-connective> ::= (SU | SR | MR | MU)  
 <constraint> ::= (\* | + | ? | "P")

Furthermore, we develop specifications for representing domain ontologies, problem-solving ontologies. With respect to domain ontologies, for instance, a domain ontology Closing Price can be informally expressed as:

```
;; Definition of Closing Price in DO
(Domain Closing_Price LI)
(substitute_to Closing_Price (Close_Price Daily_Price SI))
```

It can be formally expressed in terms of the DL as:

```
Closing_Price ::= Closing_Price (MU Close_Price Daily_Price ) (AT-MOST 1 Stock_Code)
(OO Float ?)
```

It means that there is at most one value of the closing price for a stock in a day; if there is a value existing, then its value is in float.

PSO consists of ontologies for tasks, methods, business logics and resources. The relationships among them are as follows. A task is fulfilled by some methods. A method is instantiated into some business logics, and supported by some resources. In most conditions, a task is divided into multiple sub-tasks; these sub-tasks are satisfied by some alternative methods. Correspondingly, these alternative methods are implemented by alternative business logics and relevant resources. PSO diagram as shown in Figure 6 is developed to present these items and their relationships. This diagram should be instantiated to include sets of PSO items from task, method, BL and resources in a real example.

To present these types of PSO items, the following specification is built based on first-order logics:

```
/*CCD entry for presenting a part of PSO item*/
<PSO item> ::= (<term> | <relationships>)*
<term> ::= (<name>, <type> [, cardinality] : <value>, <type>)*
<name_type> ::= (task | method | business logic | resource)
<relationship> ::= (<name>, <type> [,cardinality] : <value>, <type>)*
```

Furthermore, a PSO item can be presented as:

```
/*CCD entry for a complete PSO item*/
<PSO entry> ::= (<task> [,method][.business logic][.resource] | <ontocommit>)*
```

For instance, the following illustrates that a task *Register\_Algorithm* call BL *Type\_AlgoInputs* to fill in *Algo\_Name* as *MovingAverage* via the method *RegisterAgent*.

```
;; Definition of a PSO item
(Task Register_Algorithm SR)
(Method RegisterAgent SU)
(BL Type_AlgoInputs MU)
(BL Algo_Name (MovingAverage))
(Resource (AlgorithmBase SystemBase))
```

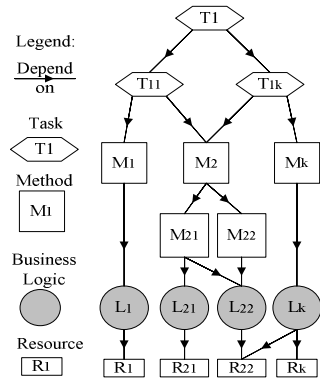


Fig. 6. PSO structure

## 5 Ontological Aggregation, Transformation, Mapping and Query

After defining and building semantic relationships and the representation of ontological items, relevant mechanisms must be developed for ontological aggregation, transformation, mapping and queries either intra or inter domains [2,3,5,6]. A key step is to build some proper ontological architecture first. An ontological architecture indicates the relationship, organization and management of the mapping and integration of ontology crossing domains. Please refer to [2,3,6] for more information about our proposed ontological architecture for DW.

Furthermore, a fundamental issue is how to aggregate, transform, map and query ontological items in the DW ontological domains. To this end, semantic rule [5] is an effective approach. Thus, we need to develop semantic rules for the aggregation of ontological items and semantic relationships, the rules for ontological transformation and mapping. These rules define what the resulting logical output should be for a given input logical combination with some semantic relationship inside. [5,6] present more information about these issues. For instance, the following rule aggregate the semantic relationship *part\_of*.

RULE 3.

$$- \forall (A \text{ AND } B), \exists B ::= \text{part\_of}(A, B) \\ \Rightarrow B, \text{ the resulting output is } B$$

Again, the following exemplifies some rules for the transformation, where  $C_i$  is an input item,  $O$ ,  $O_1$  and  $O_2$  are candidate items in the target domains.

RULE 6.

$$- \forall C_i, \exists: (\text{substitute\_to}(O, C_i) \vee \text{is\_a}(O, C_i) \vee \text{instance\_of}(O, C_i) \vee \text{part\_of}(O, C_i) \vee \text{associate\_with}(O, C_i)) \Rightarrow O, \text{ the } O \text{ is the output item}$$

## 6 Conclusions

In our previous work, we have proposed ontology-based business intelligence for improving the performance of existing commercial DW systems. This paper outlines the basic picture of ontological engineering in data warehousing. It has demonstrated mechanisms for: (i) building ontology profiles for a domain problem, (ii) defining ontological commitment and semantic relationships, (iii) presenting ontological items, and (iv) aggregating and transforming items in one domain or across domains. This actually forms the foundations for analyzing, designing and implementing ontology-based data warehouse systems in business world.

## References

- [1] Cao, L.B. et. al.: Systematic engineering in designing architecture of telecommunications business intelligence system. Proceedings of HIS'03, pp1084-1093, IOS press, 2003.
- [2] Cao, L.B. et. al.: Ontology Services-Based Information Integration in Mining Telecom Business Intelligence. Proceeding of PRICAI04, Springer Press, pp85-94, 2004.

- [3] Cao, L.B. et. al.: Integration of Business Intelligence Based on Three-Level Ontology Services. Proceedings of the 2004 IEEE/WIC/ACM International Conference on Web Intelligence, IEEE Computer Society Press, pp17-23.
- [4] Cao, L.B. et. al.: Agent Services-Based Infrastructure for Online Assessment of Trading Strategies. Proceedings of the 2004 IEEE/WIC/ACM International Conference on Intelligent Agent Technology, IEEE Computer Society Press, pp345-349.
- [5] Cao, L.B. et. al.: Ontology Transformation in Multiple Domains. 17th Australian Joint Conference on AI, G.I. Webb and Xinghuo Yu (Eds.): LNAI 3339, pp. 985–990, 2004.
- [6] Cao, L.B., Zhang, C.Z., Liu, J.M. Ontology-Based Integration of Business Intelligence. Int. J. on Web Intelligence and Agent Systems, No 4 Vol 4, 2006.
- [7] F-Trade: <http://www.f-trade.info>.