A Transparent Collaborative Integration Approach for Ad Hoc Product Data

Jingzhi Guo
Faculty of Science and Technology, University of Macau,
Av. Padre Tomás, Pereira, S.J., Taipa, Macau, Tel: +853-397 4890
Email: jzguo@umac.mo

Abstract

Product data integration is an essential issue for many e-commerce interoperable business systems. Core to this issue is how to maintain semantic consistency between heterogeneous product data that are semantically different in structure, concept and context. To resolve the issue, this paper proposes a transparent collaborative integration approach, which is built on a layered framework of messaging, structure, concept and collaboration. The key to this framework is three collaboration engines, which resolves semantic consistency issues between millions of contextual vocabularies of SMEs.

1. Introduction

Product data integration [3] is an essential issue for many e-commerce interoperable systems. Core to this issue is how to maintain semantic consistency of product data between millions of small and medium sized enterprises (SMEs) so that SMEs could understand and interoperate with each other. Consider, for example, there are two web-distributed SMEs who have individually designed their ad hoc product data on their local vocabularies. Suppose that the two SMEs are trying to collaborate with each other to sell and buy refrigerators. Now if SME_1 (buyer) encodes the refrigerator in Voc_1 as:

```
<product name = "fridge">
 <description>
  <color>silver</color>
  <price>USD560</price>
  <dimension>29 5/8"×33 1/4"×66 3/4"</dimension>
  <energyConsumption>228 kw/h/year</energyConsumption>
 </description></product>
```

and SME_2 (seller) encodes the refrigerator in Voc_2 as:

```
<product name = "freezer">
 <attribute name = "CLR">silver</attribute>
 <attribute name = "SIZE" type = "compound">
  <attribute name = "W">75.5 cm</attribute>
  <attribute name = "L">84.46 cm</attribute>
  <attribute name = "H">169.55 cm</attribute> </attribute>
 <attribute name = "PRC">550</attribute></product>.
```

The interoperation between Voc_1 and Voc_2 immediately becomes an issue. It is clear that the two pieces of ad hoc product data conflict in *syntax* (e.g. different message structures in schema level), the *semantics* (e.g. term conflicts in product, attribute and value naming) and *context* (e.g. different conceptualization such as usd560 and 550). Without proper integration solutions, the two SMEs may have to manually find each other, physically communicate in paper, and face-to-face exchange information.

In response to the above issue, three relevant strategies can be adopted: *standardization strategy* [1], *mediation strategy* [6] and *collaboration strategy* [4]. While these strategies have different pros and cons, this paper adopts collaborative strategy to integrate millions of distributed SMEs that ad hoc design their product data. With this strategy, we propose a *transparent collaborative integration* (TCI) approach to solve the issue, which aims to collaboratively and transparently design common product data that can be used to map heterogeneous ad hoc product data between semantically different SMEs.

The remainder of this paper is arranged in the following. Section 2 proposes a novel TCI framework. Section 3 describes the collaboration engines that solve the collaboration issue. The final section concludes the paper and provides the future work.

2. TCI Framework

The novel TCI framework is shown in Fig. 1, which consists of four components: transparent messaging (TM), structure repository (SR), concept repository (CR), and collaboration engines (CEs). The design thought comes from open protocol (e.g. XML SOAP), semiotics (e.g. dyadic sign model [7]:67), classifier-based product catalogues (e.g. www.UNSPSC.org), common information space (e.g. [2]) and collaborative editing systems.

Messaging layer. The TM in messaging layer is responsible for sending and receiving collaborative messages. Its need is that SMEs are web-distributed. Each may have different platform of hardware and software. Thus, a transparent messaging mechanism for message exchange is required. In TCI approach, we adopt SOAP as the mandatory message envelope for exchanging collaborators' collaborative operation documents and SME users' reified product documents. This enables collaborative operations and exchanging information to be encapsulated in SOAP body for cross-platform communication.

Structure layer. The SR provides structures for collaborative messages. *Structure* is a container of meaningful concepts. It is similar to schema, but it itself has no meaning and is only a data representation structure. With-

Proceedings of the 8th IEEE International Conference on E-Commerce Technology and the 3rd IEEE International Conference on Enterprise Computing, E-Commerce, and E-Services (CEC/EEE'06)
0-7695-2511-3/06 $20.00 © 2006 IEEE

out specific concepts conveyed in it, it is meaningless and non-useful. Structure is important for interoperation between schematically different SMEs. If there is no structure to be shared or mapped between SMEs (e.g. <*name id*=“”> and <*concept iid*=“” *annotation*=“”/>), meaningful concepts have nowhere to be conveyed and mediated between parties.

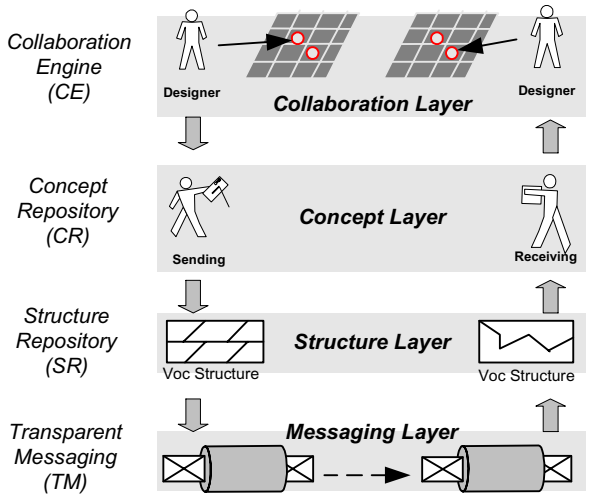

Fig. 1: A transparent collaborative integration framework

In TCI framework, we adopt an *XML Business Information* (XBI) specification (the earlier version is in [5]) as the structure for product information storage such that:

```
<!ELEMENT voc (c*)>
<!ATTLIST voc an CDATA #REQUIRED  iid ID #REQUIRED>

<!ELEMENT c (c*,val?)>
<!ATTLIST c  iid ID #REQUIRED an CDATA #REQUIRED
          co CDATA #REQUIRED  g CDATA #REQUIRED>

<!ELEMENT val (#PCDATA)>
<!ATTLIST val  cvt CDATA #IMPLIED
               dt CDATA #REQUIRED>.
```

where structure symbols are *voc* (product vocabulary), *c* (concept), *an* (annotation defining *voc* or *c*), *iid* (identifier identifying concept *c*), *co* (connotation defining the number of sub-concepts), *g* (concept group), *val* (value defining reification of concept), *cvt* (conversion function converting concept from one context of SME to another, and *dt* (data type of reified concept value).

XBI structure constrains that all product information must be structured in XBI format so that collaboration operations between collaborative parties (i.e. both SMEs and vocabulary providers) can transparently operate on the structurally same product documents.

Concept layer. The CR is a concept storage component, recording the *concepts* [5] agreed between collaborative parties into a *conceptualized* XBI vocabulary (i.e. a result of incremental instantiation of XBI DTD without reifying value structure *val*). A conceptualized XBI vocabulary is a set of *concepts* shown below:

```
<voc iid="c" an="CONEX Common Product Vocabulary">
 ……
  <c iid="c.52.14.15.1" an="Refrigerators" g="product" co="*">
   <c iid="c.52.14.15.1.2" an="price" g="attribute" co="3">
    <c iid="c.52.14.15.1.2.1" an="currency" g="Scalar" co="0"/>
    <c iid="c.52.14.15.1.2.2" an="value" g="Value" c="0"/>
    <c iid="c.52.14.15.1.2.3" an="quantity" g="Unit" co="0"/></c>
   <c iid="c.52.14.15.1.3" an="colour" g="Constant" co="0"/>
   …..
  </c></c></c></c></c></voc>.
```

The example shows a category of concepts mutually agreed and recorded by collaboration mechanism as a common product vocabulary (*ComVoc*) between vocabulary providers. This *ComVoc* is the *precondition* of further collaboration between vocabulary providers and SMEs to produce local product vocabulary (*LocVoc*) in SMEs.

Collaboration layer. The CE component is a collaboration mechanism enabling collaboration between product data designers. It includes collaborative engines (CEs), which are interconnected using a *peer-to-peer (P2P) and dominator-to-follower (D2F) architecture* in Fig. 2, where common vocabulary (*ComVoc*) in P2P network is *semantically replicated* (e.g. given $voc_1(firm_1)$: *1.52.14.15.1 ←refrigerator* and $voc_2(firm_2)$: *1.52.14.15.1← rèfrigèrateur*, then voc_1 and voc_2 is semantically replicated through concept identifier 1.52.14.15.1), and local vocabularies (*LocVoc*) in D2F network are *semantic subsets* of *ComVoc* in heterogeneous annotations (e.g. given $voc_1(firm_1)$: *1.52.14.15.1←refrigerator* and *1.52.14.15.2← ovens*, and $voc_3(firm_3)$: *FH345←fridge* and *map* (*1.52.14.15.1*, *FH345*), then voc_3 is a semantic subset of voc_1).

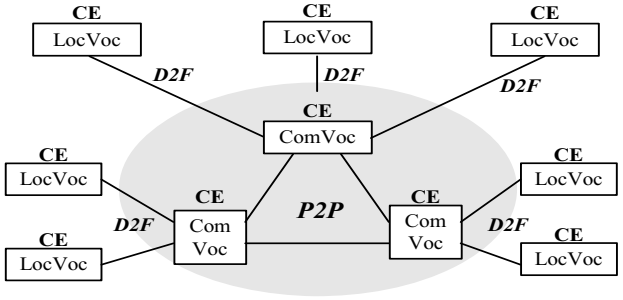

Fig. 2: Peer-to-peer and dominator-to-follower architecture

In P2P/D2F architecture, *ComVoc* and *LocVoc* are *conceptualized* from XBI DTD. *ComVoc* records P2P collaboration outcome between *ComVoc* designers (often from vocabulary providers). *LocVocs* records D2F collaboration results between *LocVoc* designers (i.e. the *followers* from SMEs) and *ComVoc* designers (i.e. *dominators* from vocabulary providers). This architecture provides a collaborative framework through collaboration engines for linking semantically heterogeneous ad hoc product data of different SMEs such that: $\text{LocVoc}_1 \leftrightarrow \text{ComVoc}_1 \leftrightarrow \text{ComVoc}_2 \leftrightarrow \text{LocVoc}_2$. For example, *c*(*p112*,

fridge) ↔ *c*(*1.52.14.15.1*, *refrigerator*) ↔ (*1.52.14.15.1*, *电冰箱*) ↔ (*x34*, *雪柜*).

Nevertheless, to design such a collaboration mechanism, some important design issues need to be resolved.

Issue 1: How to maintain semantic consistency between different *ComVocs* where concept identifiers may conflict? For example, $ComVoc_1$ generates a child concept identifier *iid*="*1.52.14.15.1.1*" and assigns a concept definition "*colour*" under the parent concept "*fridge*" identified by *1.52.14.15.1*. Simultaneously, $ComVoc_2$ generates a child concept identifier *iid*="*1.52.14.15.1.2*" and assigns a concept definition "*color*" under the parent concept "fridge" identified by *1.52.14.15.1*. The result is that two identifiers 1.52.14.15.1.1 and 1.52.14.15.1.2 of $ComVoc_1$ and $ComVoc_2$ have assigned same concept definitions and thus semantically conflict.

Issue 2: How to handle the offline issue of certain CE of *ComVoc*? For example, at *time*=1, both Voc_1 and Voc_2 have *<c iid=1.52.14.15.1 an =* "*refrigerator*"/> while at *time*=2, Voc_2 is in offline status but Voc_1 has changed to *<c iid = 1.52.14.15.2* an = "*refrigerator*"/>. This issue causes the semantic inconsistency between the offline *ComVoc* and online *ComVoc*.

Issue 3: How to consistently map contextual vocabularies (*LocVocs*) onto a common vocabulary (*ComVoc*), e.g. *LocVoc*: *<c iid*="*DH55*" *an =* "*fridge*"/> vs. *ComVoc*: *<c iid*="*1.52.14.15.1*" *an*="*refrigerator*"/>? This needs a solution that can map *LocVocs* onto *comVoc* and personalize local concept identifier of *LocVocs*.

3. Collaboration Engines

To resolve the above issues, we reconstruct P2P/D2F architecture into a new centrally-managed P2P/D2F architecture shown in Fig. 3 with three different collaboration engines: universal collaboration engine (UCE), common collaboration engine (CCE) and local collaboration engine (LCE), where they respectively resolve the issues of concept identifier conflicts, offline inconsistency, and contextual local vocabularies.

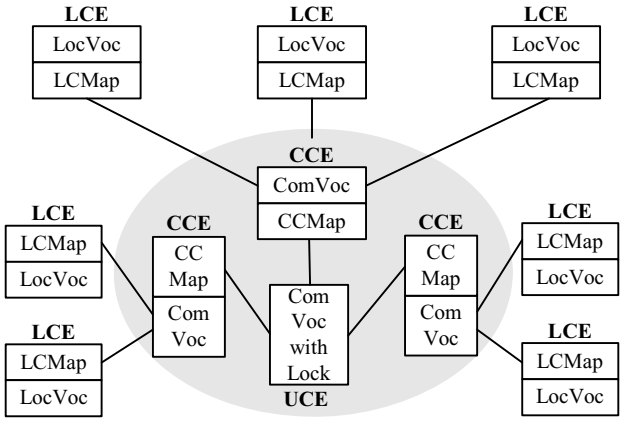

Fig. 3: Centrally-managed P2P/D2F architecture

In this new architecture, the three types of connected collaborative engines build the collaboration mechanism of TCI framework and enable semantically heterogeneous product information interoperable. In this mechanism, the CCEs work together to form a P2P collaborative network through the *common-to-common maps* (CCMap) and the UCE coordination. A collection of LCEs connects to a certain CCE to create a D2F collaboration network through the *local-to-common maps* (LCMap).

Universal collaboration engine. The UCE is designed to resolve concept identifier conflicts through coordinating CCEs. As described before, multiple *ComVocs* are semantically replicated, which means each CCE has a same copy of *ComVoc* identifiers. When multiple CCEs simultaneously work on a certain concept of vocabulary tree, concept identifier conflicts happen. To resolve the issue, we introduce a *concept node locking mechanism*, shown as in Fig. 4, in UCE. This mechanism creates a new *ComVoc* with an additional lock attribute on each concept such that:

<c iid = "" an = "" lock = "on | off"/>.

This special vocabulary is centrally managed as a dedicated lock tree in a separate web server, where all CCEs can access. When a CCE needs to create a child concept, it issues a lock notice to the dedicated lock tree where the corresponding sibling concepts are marked as *locked*. The *locked* means that when the concept node is locked, other CCEs cannot create the same nodes in their own CCEs. However, they can still issue locks under the existing child nodes of the locked nodes to create the grandchildren concepts by issuing locks on grandchildren concepts, shown as in Fig. 4 (details will be presented elsewhere).

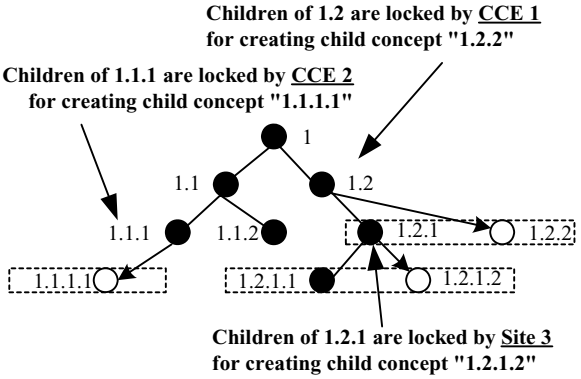

Fig. 4: Concept node locking mechanism

Simply, an existing node can be issued a set of locks on all its children, which do not affect lower level descendant nodes. The granularity of the *concept node lock* is relatively small. The reason why the sibling nodes of the node being created should be locked is that multiple child nodes may be created but be assigned the same annotated meanings. This will create redundant concepts under the same parent concept and cause semantic incon-

Proceedings of the 8th IEEE International Conference on E-Commerce Technology and the 3rd IEEE International Conference on Enterprise Computing, E-Commerce, and E-Services (CEC/EEE'06)
0-7695-2511-3/06 $20.00 © 2006 IEEE

sistency in concept classification, which will further develop many heterogeneous sub concept classifications.

Common collaboration engine. A CCE of a P2P network can be offline in whatever reason. When it is offline, a previously recorded concept identifier may be changed or a previously recorded concept annotation defining a concept identifier may be changed. This causes the CCE offline issue. A CCMap is devised to support CCE and let it avoid the semantic inconsistency between the offline CCE and online UCE. A CCMap is a concept map between a *ComVoc* concept of CCE and a *ComVoc* concept of UCE in terms of a outgoing buffer (*OB*) and a incoming buffer (*IB*) such that:

```
<ccmap>
  <ob oplist = “”/> <!-- OB at UCE -->
  <ib oplist = “”/> <!-- IB at CCE -->
</ccmap>.
```

where the *oplist* is an operation list such that *oplist* = $[op_k(iid_k, an_k), \ldots, op_1(iid_1, an_1)]$ which is a history buffer of editing operations on UCE after the CCE goes offline.

When the CCE goes online, the *OB* of UCE flushes to *IB* of CCE to synchronize. In our current design, CCE is not allowed to edit offline. Thus, the CCMAP is a one-way buffering mechanism. The design of CCMap not only resolves the offline issue of CCE but also provides the flexibility of editing product vocabulary.

Local collaboration engine. The heterogeneous context issue between *ComVoc* and *LocVoc* is resolved in LCE of D2F network through LCMAP, which maps *LocVoc* onto *ComVoc*. The *LCMap* consists of a local concept identifier (*locIid*) and a common concept identifier (*comIid*) such that:

LCMap(locIid, comIid).

When an LCE creates a new ad hoc concept on *LocVoc* based on *ComVoc*, it at the same time creates an *LCMap*(*locIid*, *comIid*) between *LocVoc*(*locIid*←*locAn*) and *ComVoc*(*comIid*←*comAn*). With this map, both CCE and LCE can freely edit their individual *ComVoc* and *LocVoc* without causing the semantic consistency issue. The simple concept identifier mapping is powerful. It enables ad hoc product data semantically integrated. For example, given <*c iid*=“123” *an*=“*fridge*”/> and <*c iid*= “1.52.14.15.1” *an*=“*refrigerator*”>, a map <*lcmap locIid* = “123” *comIid*= “1.52.14.15.1”> makes them integrated.

If the one-way buffering mechanism of *CCMap* applied in *LCMap*, the LCE can further asynchronously collaborate with the CCE such that CCE maintains OB while LCE maintains IB.

4. Conclusion

This paper has proposed a transparent collaborative integration (TCI) approach to maintaining semantic consistency between the multiple common product vocabularies in a P2P network and between multiple local product vocabularies and a certain common product vocabulary in a D2F network. The approach first presents a TCI framework consisted of the layers of messaging, structure, concept and collaboration. This framework is built on a centrally-managed P2P and D2F architecture where the transparent messaging is provided through open SOAP protocol. The reusable structure of product vocabulary adopts an XBI specification. Collaborative product concepts are recorded in conceptualized XBI documents, which ensure collaboration results to be properly stored. The semantic consistency maintenance is resolved through three collaboration engines: universal collaboration engine, common collaboration engine and local collaboration engine.

Comparing with approaches of standardization and ontology mediation, TCI approach is collaboration-based, which is novel and can support the integration of millions of SMEs’ ad hoc vocabularies. This ability increases the scalability of product data integration systems.

Future work of this paper is to describe the transparent collaboration operations that can be universally and dynamically used in different collaboration engines.

5. Acknowledgement

Thanks to the four anonymous reviewers for their insightful comments that help improve the presentation.

6. Reference

[1] Arpinar, S., and A. Dogac, “Provision of Market Services for eCo Compliant Electronic Marketplaces”, *ACM SIGMOD Record*, Vol. 29, No.3, 2000, pp.24-30.

[2] Bannon, L. and Bϕdker, S., “Constructing Common Information Spaces”, in: *Proceedings of ECSCW’97*, 7-11 September, 1997, Lancaster, UK.

[3] Fensel, D., Ding, Y., Omelayenko, B., Schulten, E., Botquin, G., Brown, M. and A. Flett, “Product Data Integration in B2B E-Commerce”, *IEEE Intelligent Systems* 16(4), 2001, pp.54-59.

[4] Guo, J. and C. Sun, “Collaborative Product Representation for Emergent Electronic Marketplace”, in: *Proc. of 16th Bled Electronic Conf.*, Bled, Slovenia, June 9-11, 2003, pp.847-859.

[5] Guo, J. and C. Sun, “Context Representation, Transformation and Comparison for Ad Hoc Product Data Exchange”, in: Proc. of *ACM DocEng’03*, Grenoble, France, November 20-22, 2003, pp.121-130.

[6] Omelayenko. B. and D. Fensel, “A Two-Layered Integration Approach for Product Information in B2B E-Commerce”, in: *Proc. of the 2nd Int’l Conf. on Electronic Commerce and Web Technologies*, LNCS 2115, Springer-Verlag Berlin, 2001, pp.226-239.

[7] Saussure, F, *Course in General Linguistics*, McGraw-Hill Books Company, 1966.