Context Representation of Product Data

JINGZHI GUO AND CHENGZHENG SUN

This paper investigates the context representation of *ad hoc* product data that distribute in different semantic communities especially in small and medium sized enterprises through the analysis of the problems in extracting, mediating and comparing product contexts. The problem analysis leads to a proposal for a novel context representation model that separates the representation process into two steps: transform irregular local product definitions into canonical local product representations and map canonical local product representations onto common product representations. This model serves as the foundation to further compare product contexts between different semantic communities.

Categories and Subject Descriptors: H.5 [**Information Interfaces and Presentation**]: Group and Organization Interfaces – *Web-based interaction*; H3.5 [**Information Storage and Retrieval**]: Online Information Services – *Sharing Data*; *Web-based services*
General Terms: context representation, product data integration, product catalogue, electronic commerce

1. INTRODUCTION

Product data integration is an essential issue for business-to-business interoperation to form global electronic marketplace [1][2][8]. It deals with how to extract product data from rough sources that are heterogeneously represented in different "semantic communities" [11], and how to establish semantic similarities and equivalences between different product data to create mappings (c.f. [7]). For these two aspects, the former aims to find a solution to representing the dynamic contexts of product data in different semantic communities and placing them in a consistent product representation model. The latter focuses on how to compare these contexts and create mappings based on a common product concept model or domain vocabularies such as ontologies. In this paper, we only focus on the first aspect of the issue.

The context representation of heterogeneous product data is important and is a means of capturing the semantics of locally defined product data. It explicates heterogeneously structured, differently annotated and implicit semantics. It places the contexts of all heterogeneous product data in a consistent product representation model so that all captured semantics can be compared for integration into the other domains of product vocabularies.

Current discussions about the context representation of data mainly appear in the database area such as the works of Goh et al [3] and Kashyap et al [7]. In product data integration area, this issue is not yet emphasized and largely neglected. In the database area, the inadequacy of purely structural representations has been found that the ability to represent the structure of an object does not help capture the real-world semantics of the objects. The structural equivalence is necessary but not sufficient to determine the semantic similarity or equivalence [7]. The nature of product data is similar to those that are discussed in the database area. However, it is even more challenging due to the very large number of *ad hoc* product data representations and their fast evolution [8]. Specifically, heterogeneous product data exist in many types of data sources listed in the following:

Address: School of Computing and Information Technology, Griffith University, Nathan, QLD 4111 Australia. Email: {J.Guo, C.Sun}@cit.gu.edu.au

- Numerous heterogeneous de facto industrial product standards such as xCBL (www.xcbl.org), cXML (www.cxml.org) and ebXML (www.ebxml.org) that prevent business interoperations [1][8][10].
- Several heterogeneous international product classification standards such as UNSPSC (www.unspsc.com) and eCl@ss (www.eclass-online.com) present different guidelines of classifying products [12].
- Millions of *ad hoc* enterprise-wide product data representations that still increase in an exponential way. These representations generally exist in small and medium sized enterprises (SMEs) that are financially and technically impossible to join any existing product standards if requiring large amount of reengineering work of their product data representations.

The task of product data integration is to integrate the above heterogeneous representations for business interoperation. However, the integration of any type of above data is a great challenge [5]. In this paper, we only focus on how to represent the contexts of *ad hoc* enterprise-wide product data for later integration research.

In the remaining part of this paper, we will investigate and design a context representation model for capturing semantics. In section 2, the definition of context and problems are discussed. Section 3 proposes a novel context representation model to solve context representation problems. Section 4 discusses two additional issues that are important to the proposed model. Section 5 concludes the paper, lists the important contributions, gives the research implications and suggests immediate future work.

2. CONTEXT AND ITS PROBLEMS

2.1. Context Definition

In multidatabases, "the context is the key component in capturing the semantics related to an object's definition and its relationship to other objects" [7]. In product data representations, the *context* is the semantics definition of a product related to local product representations in a semantic community and common product representations in a product catalogue provider. A *semantic community* is a group of people, an enterprise or multiple enterprises who share the same perspectives to interpret the same products and represent their products in the identical ways (c.f. [11]). Representing the context of a product is to extract the semantic definition of a product in a semantic community and to expose it to other semantic communities disregarding how this product is schematically defined.

The differences between the contexts in the database research such as in semPro [7] and that defined in this paper are: the former "refers to the context in which a particular semantic similarity holds between two objects" [7], while the latter refers to a particular similarity between a local product representation and a common product representation. The redefinition of context in this paper considers the fact that there are millions of enterprises and thus there are millions of product definitions. A context should be represented disregarding how products are locally defined in structures and data store models. This strategy is to avoid the large number of comparisons during semantic comparisons at the time of data integration. In another word, the different contexts should be compared indirectly through a set of common product representations. By this strategy, we reduce the number of mappings and convert the many-to-many mapping relationship to one-to-one mapping relationship.

2.2. Problems in Extracting Contexts

The sources of the contexts of product data exist in different semantic communities. Each semantic community may represent their product data in their own formats though the actual semantics are possibly the same as other semantic communities. Often product data representation formats can be categorized into relational table record(s), XML node(s) or files, or *ad hoc* web data such as shown in Fig. 1 (a). The multiple sources of representations increase the difficulties to represent contexts. In fact, it is impossible to compare semantics if the semantics are implicitly notated in different irregular representations. To solve this problem, this paper suggests transforming the implicit and irregular product data into canonical local product representations such as shown in Fig. 1 (b).

2.3. Problems in Mediating Contexts

In multidatabases, the contexts are mediated by the standard terms. For example, semPro uses ontology [7]. However, using product ontology as the mediating vocabularies has a precondition: either the product data sources are built according to the shared product ontologies, or at least the product definitions can commit to the ontologies whenever they are later adopted. Obviously, in the former case, there should be pre-existed shared ontologies that make the product data sources understand them. In the latter case, if the ontologies are later plugged in for mediation and the systems use intelligent agents to extract the sources to map ontologies, then many significant problems will happen such as "an inadequate representation format of the information and high irregularities in the information's layout", "incompleteness" and "false values" [2]. In business reality, most product data exist in SMEs and their product data formats are *ad hoc*, thus at most the latter ontology mediating approach can apply, which means that we are facing above listed problems in mediating contexts.

To solve the mediating problems, we provide common product representations as runtime semantics reference systems such that when product contexts are extracted from the data sources, these product contexts can be compared against a set of common product representations to check whether the contexts can be mediated.

2.4. Problems in Comparing Contexts

In various product data sources, a product is locally defined according to the local systems resources and preferences of a semantic community. For example, in Fig. 1, product A in domain A defines a refrigerator in several relational table records, product B in domain B defines a refrigerator in an XML file and product C in domain C defines a refrigerator in an *ad hoc* web page. These different product definitions are irregular and conflict with each other. Specifically, the semantic conflicts can be listed in the following:

- *Product classification conflicts* arise from the same product classified in different classified levels or different branches of a classification, which results in different concept denotation and connotation. Semantics may be crossing. For example, one may classify "refrigerator" in (home electronics ((refrigerator), (Oven))) while another may classify it in (refrigeration systems ((domestic refrigerator), (industrial refrigerator))).
- *Product definition conflicts* arise from the unmatched product identifier and its definition or annotation. For instance, an identifier "1112" may have been assigned "cooling systems" in one place and "drilling systems" in another place. In another case, "refrigerator" is assigned "2222" in one place and "fridge" is assigned "3345" in another place though both actually mean the same. In Fig. 1, product A, B and C use the "model" type to identify the products, and product names are the descriptions of the model types. Obviously, they are not interoperable.
- *Product structure conflicts* arise from a same product defined by the different number of and kinds of attributes. For example, a television is either defined as (color,

size, weight, thickness) or defined as (color, size, screen type). In Fig. 1, product A, B and C have different attribute types and attribute numbers.

- *Attribute description conflicts* are similar to product definition conflicts (e.g. definition of "dimension" for refrigerator may be differently defined as "measurement"). In addition, attribute description may be implicit. For example, in product C of Fig. 1, "color" attribute is implicit to mark "silver" on the page.
- *Measurement conflicts* arise from the different scales or units of the same attribute value. For example, gloves are sold either in piece or in pair. Sometimes an attribute value can have several implicit measurement standards, for example, "43" and "9" roughly mean the same "size" of shoes when applying different measurement systems. In Fig. 1, "freezing capacity" in product B and product C is different. The former uses "litre" while the latter uses "kg/24h".
- *Attribute value definition conflicts* arise from the different definitions of an attribute value. For instance, in Fig. 1, "energy consumption" of product B and C have different value definition. Product B implicitly defines "kwh per day" while product C explicitly defines "kwh per year".

Semantic conflicts arisen from the definition structures indicate that their contexts are schematically non-comparable. To capture the semantics, we compare a set of canonical local product representations with a set of common product representations. The purpose is to build the product contexts that have the common ground for comparison.

Product A in domain A - Relational Database

ID	Name	Model	Dimension	Weight	Color	Description
5	Refrigerator	HTQ18JAAWW	d255	w132	white	full-size, frost free, adjustable Shelves

Dimension:

ID	Width	Depth	Height
d255	29 5/8"	33 1/4"	66 3/4"

Weight:

ID	Unit	Value
w132	lbs	238

Product B in domain B - XML File

```
<refrigerator   model = "BCD-202WG">
  <name> fruit refrigerator  </name>
  <price>  630 </price>
  <price currency = "Australia dollar">   630  </price>
  <gross_capacity unit = "litre"> 202  </gross_capacity>
  <fresh_capacity unit = "litre"> 122  </fresh_capacity>
  <freezing_capacity unit = "litre"> 80  </freezing_capacity>
  <energy_consumption unit = "kwh"> 0.97  </energy_consumption>
</refrigerator>
```

Product C in domain C - Ad-hoc Format

REFRIGERATOR
Fresh food capacity: 247 litres
Automatic defrost
3 adjustable glass shelves
1 fruit and vegetable container
1 covered storage box
1 chrome wine rack

Gross capacity: 271 litres
Tropicalized compressor
Adjustable thermostat
Energy efficiency class: A
Energy consumption: 288 Kw/h per year
Climatic class: T
Freezing capacity: 2kg/24h
Thaw time: 12h

silver **FAB2ÅZ3**

(a)

Local Product Representation of Product C

```
LocRep(LD_A) = PR(fab2az3, refrigerator, PS)
PS = AT(capacity, defrost, shlves, container,  storage box,
        wine rack, compressor, thermostat, energy efficiency class,
        energy consumption, climatic class, thaw time, color)
AT(capacity) = (fab2az3-capacity, capacity, AS)
 AS(fab2az3-capacity) = VT(fresh food, gross, freezing)
  VT(fresh food) = (fab2az3-capacity-fresh_food, fresh food, VS)
   VS(fresh food) = (unit, integer)
     Assign(unit) = litre
      Assign(value) = 247
   VT(gross) = (fab2az3-capacity-gross, gross, VS)
    VS(gross) = (unit, integer)
      Assign(unit) = litre
       Assign(value) = 271
   VT(freezing) = (fab2az3-capacity-freezing, freezing, VS)
     VS(freezing) = VT_2(value, scale)
       VT_2(value) = (fab2az3-capacity-freezing-kg, kg, 2)
       VT_2(scale) = (fab2az3-capacity-freezing-h, hour, 24)
AT(defrost) = (fab2az3-defrost, defrost, automatic)
 AT(shelves) = (fab2az3-shelves, shelves, AS)
  AS(fab2az3-shelves) = VT(quantity, adjustability, material)
    VT(quantity) = (fab2az3-defrost-quantity, quantity, 3)
    VT(adjustability) = (fab2az3-defrost-adjust, adustability, yes)
     VT(material) = (fab2az3-defrost-material, material, glass)
....  .......  ........  .........
(Omitted )
```

(b)

Fig. 1: Local Product Definition and Local Product Representation

3. CONTEXT REPRESENTATION MODEL

In this section, we will present a context representation model to capture the semantics of product data to solve the above problems. We will discuss the local product representations and common product representations. We will describe how to map the local product representations onto common product representations to represent the contexts.

3.1. Local Product Representations

To solve the problem of context extracting, this section initiates a novel work and proposes a model of local product representations to transform irregular local product definitions into canonical local product representations.

Given an irregular local product definition *LD*, the canonical *local product representation* is defined by an annotated nested triple [4].

$$LocRep(LD_i)=_{def}(PC_i, PA_i, PS_i(\sum AT_{ij}(AC_{ij}, AA_{ij}, AS_{ij}(\sum VT_{ijk}(VC_{ijk}, VA_{ijk}, VS_{ijk})))))$$

where PC_i is an identifier of LD_i, PA_i is the annotated product definition of LD_i, PS_i is the product structure of LD_i that contains a set of attribute types AT, AT_{ij} is an attribute type LD_i, AC_{ij} is the attribute type identifier of AT_{ij}, AA_{ij} is the annotated attribute definition of AT_{ij}, AS_{ij} is the attribute structure of AT_{ij} that contains a set of value types VT, VT_{ijk} is a value type of AT_{ij}, VC_{ijk} is the value type identifier of VT_{ijk}, VA_{ijk} is the annotated attribute definition of VT_{ijk}, and VS_{ijk} is the value of VT_{ijk}.

- The outer layer of (PC_i, PA_i, PS_i) denotes how products are defined in *product type level*. PC identifies local products, PA defines the denotations/extensions of LD and PS defines the connotations/intensions of LD.
- The middle layer of (AC_{ij}, AA_{ij}, AS_{ij}) denotes how attributes of LD are defined in *attribute type level*. AC identifies attributes of AT, AA defines the denotations/extensions of AT and AS defines the connotations/intensions of AT.
- The inner layer of (VC_{ijk}, VA_{ijk}, VS_{ijk}) denotes how values of AT are defined in *value type level*. VC identifies value types of VT, VA defines the denotations/extensions of VT and VS defines the connotations/intensions of VT. VS are generally assigned as real values that can be dynamically changed. However, if necessary, VS can expand into more levels in structure.

In Fig. 1 (b), we illustrate the transformation result of product C. The transformation process constitutes a set of visual transformation operations. How to transform is an independent research topic and will be discussed in another paper. Nevertheless, the general steps of transformation are suggested to include (1) devising a set of visualized operations to retrieve the data contained in irregular local product definitions, (2) normalizing the retrieved data to present a set of normalized concepts, and (3) converting the normalized concepts into locRep formed concepts. By LocRep, we generate canonical local product representations ready for further context representations. The assumption of LocRep is that each local semantic community knows how to reformulate their irregular definitions according to above three steps, in that the definitions are given by themselves.

3.2. Common Product Representation

The purpose of local product representations is to make the irregular local product definitions canonical and to make the semantics explicit. However, the canonical local product representations are still local because each representation still uses locally agreed terms to express semantics. To have them global to the world, common product terms are needed for the mapping. One solution is to adopt existing product ontologies. However, due to the listed mediating problems discussed in section 2.3, the terms defined in product ontologies might not include many fast evolving local terms. For this reason, we adopt a set of common product representations as the reference of product semantics. This set of representations maps onto the canonical local product representations. A *common product representation* is the product semantics given by a product catalogue provider. It has several features: (1) the product concept is neutral and calculable; (2) its semantics is given by annotation to restrict its denotation; (3) its structure is variable, evolvable, and is given by its internal nested structures, which connotes the product concept.

The *common product representation* ComRep(CP_i) is defined as by an annotated nested triple [4][6].

$$ComRep(CD_i)=(PC_i, PA_i, PS_i(\Sigma AT_{ij}(AC_{ij}, AA_{ij}, AS_{ij}(\Sigma VT_{ijk}(VC_{ijk}, VA_{ijk}, VS_{ijk})))))$$

where CD_i is a common product definition given by catalogue providers, and is structurally the same as LocRep but in the "public" product domain.

The specific differences from LocRep are: (a) the definition domain of PC where PC is a set of calculable vector tree defined as PC=(1, 2i, 3i, …, ki) [4]. This vector tree simulates a product catalogue tree to provide standard terms of vocabulary with root $PC_{(1)}$ whose descendants are $PC_{(1, 1)}$, $PC_{(1,2i)}$ and $PC_{(1, 2i, 3i, …, ki)}$. If a PC is (1, 3, 5), it means the fifth child of the third child of the root and simply expressed as 1.3.5. (b) AC or VC is simply denoted as a sequential integer. An AC(5) of $PC_{1.3.5}$ is denoted as 1.3.5-5, and a VC(3) of AC(5) is denoted as 1.3.5-5-3. In complicated situations such that many levels are required to represent a product, a bottom level of a product representation can be denoted as 1.3.5-5-3-,…, -i. (c) PA of common products are standard annotations that can be similar to the product definitions of UNSPSC. For example, if we adopt UNSPSC as reference semantics, then we can convert 52141501 of UNSPSC into PC(1, 52, 14, 15,1), then PA of PC_1 is annotated as the root "UNSPSC", PA of $PC_{1.52}$ is annotated as "Domestic Appliances and Supplies and Consumer Electronic Products", PA of $PC_{1.52.14}$ is annotated as "domestic appliances", PA of $PC_{1.52.14.15}$ is annotated as "domestic kitchen appliances", and PA of $PC_{1.52.14.15.1}$ is annotated as "domestic refrigerator". If the "domestic refrigerator" has public attributes of "model", "dimension" and "capacity", then the "model" may be denoted as 1.52.14.15.1-1, the "dimension" may be denoted as 1.52.14.15.1-2, and the "capacity" is 1.52.14.15.1-3. The same is to the lower levels of the product. However, whenever an annotation standard is fixed, it cannot be changed.

The integration of this new standard with other standards and product ontologies can apply the most of the mapping technologies, for instance [9]. Nevertheless, it is worth to notice that the selection of a mapping technology is dependent on the "exactness" and "accuracy" requirements [5]. If exactness and accuracy is an important issue, using ComRep as a standard generally will meet the requirements in that the semantics is explicitly and unambiguously defined. ComRep is a multi-layer product concept representation mechanism. Each layer defines the concepts in annotations (denotations) and structures (connotations). Therefore, the accuracy is guaranteed. If customers require full accuracy such as placing orders, the mapping of LocRep onto ComRep can be exact in each layer in that all unmatched products can be filtered out by numerical concept comparisons. For example, comparing (1.52.14.15.1, refrigerator) with (1.52.14.15.1, fridge) ensures that "refrigerator" is semantically equal to "fridge".

In this paper, comRep is designed based on the concept exchange approach discussed in [5]. This approach guarantees that comRep can smoothly interoperate with the data provided by both public product catalogue providers and local product catalogue suppliers (locRep suppliers). To map with the data of the former, we introduce the collaborative generation of product concepts between catalogue providers (please refer to [5] for details). To map the data of the latter, we adopt a collaborative product representation approach (please refer to [4] for details).

The purpose of ComRep is to provide a mediation facility to bridge over various local product representations via context representations. It disregards how local products are structurally defined such as product A, B and C in Fig. 1. The mapping between local and common representations is the *context representation* of a local product. In Fig. 2 (a), we illustrate the context representation model. We discuss it in the following part.

3.3. Context Representation Model

The differences between LocRep and ComRep are: (a) LocRep is privately adopted in a local semantic community while ComRep is available for public. (b) LocRep adopts local *ad hoc* formats that cannot be understood by other semantic communities, while ComRep employs domain catalogue experts to design product representations according to certain standards and provide natural language annotations as integration interface for various local product catalogue designers. (c) LocRep becomes interoperable when it joins the ComRep product catalogues or all other catalogues that has already been mediated by ComRep.

Context representation is a mapping of LocRep onto ComRep. A collaboration mechanism described in [4] has stipulated how this mapping should be processed by following the interaction model between local product catalogues and common product catalogues. Applying the processed results of collaboration mechanism, semantic context of the locRep can be represented. The model of representing a context is defined in the following:

Context:LocRep→ComRep such that Context(LocRep)⊆ComRep ∪ LocRep

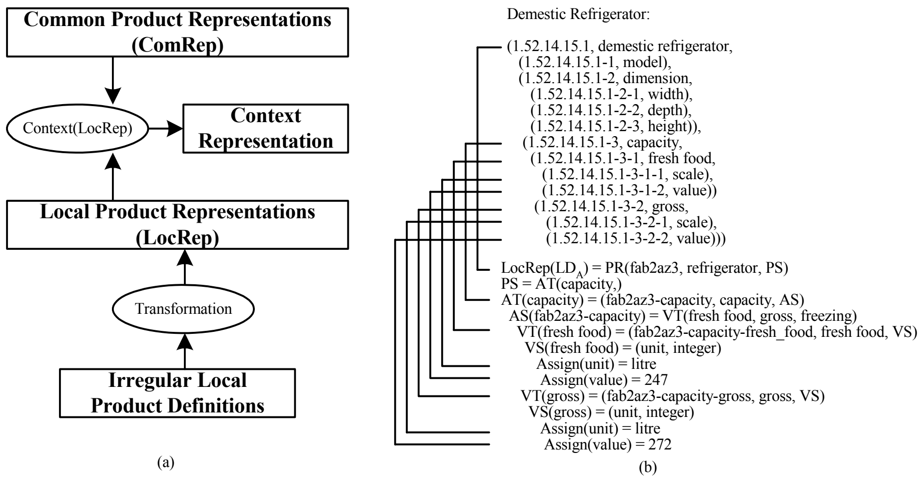

Fig. 2: Context Representation Model

Fig. 2 (b) illustrates how a canonical LocRep is mapped onto ComRep. In this example, the *Context* can be written in Figure 3:

```
(1.52.14.15.1, refrigerator,
  (1.52.14.15.1-3, capacity,
    ((1.52.14.15.1-3-1, fresh food, ((1.52.14.15.1-3-1-1, scale, litre),
                                     (1.52.14.15.1-3-1-2, value, 247))),
     (1.52.14.15.1-3-2, gross, ((1.52.14.15.1-3-2-1, scale, litre),
                                (1.52.14.15.1-3-1-2, value, 272))))))
```

Fig. 3: Example of Context Representation

Context is a partial representation of the combination of LocRep and ComRep. It replaces local product concepts such as product model type or article number by standardized concepts PC, AC and VC, but, at the same time, it retains the product descriptions and value assignment of local products. *Context* has captured the semantics of locally represented products, and it is presented to the public for product interoperation.

As for the implementation, the suggestion is that *Context* is placed in local storages. However, it can be implemented in the product catalogue providers' storages dependent on the specific requirements. Fig. 4 illustrates a high-level implementation framework of context representation.

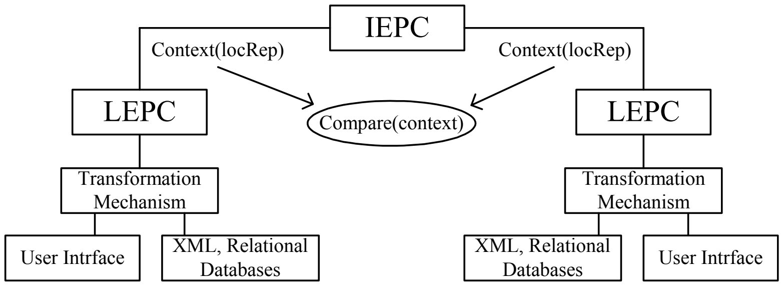

Fig. 4: A high-level implementation framework

In this framework, comRep is implemented as interoperable electronic product catalogue (IEPC) and locRep is implemented as local electronic product catalogue (LEPC) that are discussed in [4]. A user interface functions as providing instructions of how to transform irregular *ad hoc* product data of various sources stored in a local semantic communities. Transformation mechanism is responsible for transforming the retrieved irregular product definitions into canonical locRep, which is again arranged into LEPC. A Context(locRep) is the canonical representation of a local product definition that is expressed in terms of comRep. Since all Context(locRep) are expressed in comRep, they can compare with each other. If the compared results are equivalent, they can fully interoperate, that is, they are exactly matched.

4. DISCUSSION OF ADDITIONAL ISSUES

Two additional issues should be taken into considerations when representing product contexts. First, in what scope *Context* is applicable. As mentioned in section 1, the total task of product data integration is to integrate various de facto industrial standards, several product classification standards and numerous *ad hoc* enterprise product formats. ComRep aims to be a kind of flexible mediating vocabularies for numerous *ad hoc* product formats of SMEs. Therefore, to interoperate with other existing product standards, ComRep still needs to make integration efforts. In this sense, *Context* by mapping LocRep and ComRep is most applicable for representing the product contexts of *ad hoc* and irregular product representations distributed around SMEs.

In fact, focusing *Context* for SMEs is an attempt of this paper. The reason is twofold: most SMEs only have tens of or several hundred products but they take a big portion of business. *Context* provides SMEs an easy and less cost means to transform their irregular product definitions into canonical local product representations and map LocRep onto ComRep to display the product contexts. The second reason is that large companies often have hundreds or thousands of products. They either have product standards by themselves or follow certain product standards. Therefore, their product integration problems fall into the category of product standard integration that is not the focus of this paper.

Second, how to allow fast evolution of common product representations and permit dynamic local product value changes. This paper has not expanded the discussion of this issue. However, this is an important issue for maintaining a large-scale web-based product catalogue. Our initial vision to the solution of this problem is the adoption of the collaboration between local product catalogue designers and common product catalogue providers [4]. When new terms are required, two alternatives are given to the local product catalogue designers: submitting new term classification requirements to common product catalogue providers by paid services, or extending the product or attribute subtypes by themselves. The first alternative matches the sprits of *Context* that provides exactness and accuracy for information retrieval. The second alternative makes local product designers to take the inaccuracy risks.

Fast evolution of common product concepts always has contradictions either with inaccuracy by local classification or intelligent agent classification, or with large expert labors of manual classification. This issue is a balancing problem in practice.

4. CONCLUSION

This paper has discussed the problems in representing product contexts and proposed a novel context representation model by mapping local product representations and common product representations. The contributions are:

- Clarified the problems in product context extracting, comparing and mediating of product data integration.
- Proposed a novel solution to representing the product contexts of *ad hoc* product representations especially for SMEs.

This paper has opened up a wide range of research opportunities. First, we have recognized that common product concepts that are evolvable in run-time is a complex task and therefore need a well-defined methodology for accomplishing this. The issue is core to the cost-benefit analysis in terms of accuracy and completeness. This issue largely relates to questioning whether AI technology can solve classification problem, or human expert collaboration can afford the cost of classification, or chaos from *ad hoc* classification is tolerable. Second, we recognize that local product designers should be facilitated with a kind of web interface to enable quick, correct and exact common product data use for linking their local product data. Following this paper, our immediate future work is to research the context similarities between different local product representations to build the context comparison mechanism illustrated in Fig. 4.

REFERENCE

[1] Dogac, A. and I. Cingil, A Survey and Comparison of Business-to-Business E-Commerce Frameworks, *ACM SIGecom Exchanges*, Vol. 2/No. 2, June 2001, 16-27.

[2] 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*, July/August 2001, 54-59.

[3] Goh, C., Bressan, S., Madnick, S. and M. Siegel, Context Interchange: New Features and Formalisms for the Intelligent Integration of Information, *ACM Transactions on Information Systems*, Vol. 17, No. 3, July 1999, 270-293.

[4] Guo, J. and C. Sun, Collaborative Product Representation for Emergent Electronic Marketplace, *16th Bled Electronic Commerce Conference: eTransformation*, Bled, Slovenia, June 9-11, 2003.

[5] Guo, J. and C. Sun, Concept Exchange: Constructing Interoperable Electronic Product Catalogues in an Emergent Environment, *IEEE Conference on Electronic Commerce*, Newport Beach, CA, USA, 24-27, June 2003.

[6] Guo, J. and C. Sun, Conceptual Modeling of Product Representations: A Concept-Based Approach, *submitted*.

[7] Kashyap, V. and A. Sheth, Semantic and Schematic Similarities between Database Objects: A Context-Based Approach, *The VLDB Journal* (1996) 5, 276-304.

[8] Ng, W. K., Yan, G. and E. Lim, Heterogeneous Product Description in Electronic Commerce, *ACM SIGecom Exchanges*, Vol. 2000.

[9] Omelayenko, B., Integrating Vocabularies: Discovering and Representing Vocabulary Maps, *First International Semantic Web Conference (ISWC-2002)*, June 9-12, Sardinia, Italy, 2002.

[10] Omelayenko, B. and D. Fensel, An Analysis of B2B Catalogue Integration Problems, In: *Proc. of Int'l Conf. on Enterprise Information Systems (ICEIS-2001)*, Setubal, Portugal, July 7-10, 2001.

[11] Robinson, M. and L. Bannon, Questioning Representations, *ECSCW' 91*, Amsterdam, Holland, September 1991.

[12] Schulten, E., Akkermans, H., Guarino, N., Botquin, G., Lopes, N., Dörr, M. and N. Sadeh, The E-Commerce Product Classification Challenge, *IEEE Intelligent Systems* 16 (2001).