

Transforming Heterogeneous Product Concepts through Mapping Structures

Jingzhi Guo, Chengzheng Sun and David Chen
School of Computing and Information Technology, Griffith University
Nathan, QLD 4111, Australia
{J.Guo, C.Sun, D.Chen}@cit.gu.edu.au

Abstract

An unfavourable phenomenon is observed: current electronic markets are fragmented and have formed a set of autonomously distributed product information islands. This leads to heterogeneity of product information between separated sources and makes difficult on product document interoperation. To resolve the issue and facilitate business document interoperation, this paper proposes a heterogeneous concept mapping approach. By this approach, heterogeneous product documents are transformed from one context to another context without losing any semantic information. This transformation process is supported by a heterogeneous concept transformation algorithm that includes five transformation steps: source-local context transformation, local-common context transformation, common-common context transformation, common-local context transformation and local-source context transformation.

1. Introduction

As a part of cyberworld, global electronic markets (GEM) play an important role in collaborating global business operations for generating more revenues and reducing more costs [12]. However, an unfavourable phenomenon is observed in constructing GEM: current electronic markets are fragmented and have formed a set of autonomously distributed business information islands [25]. Particularly, its effects on product information exchange in semantic data layer are that heterogeneous product data between information senders and receivers are not interoperable [9]. The cause is that each fragmented electronic market has its own semantic context of product information [10]. More generally, a market, a firm or a product information system is autonomous and is a *semantic community* [28], which presents a boundary for understanding product information.

Let us illustrate in Fig. 1, given that Firm1 sends inquiry O1 based on SEPC1 and Firm2 receives O1 based on SEPC2. It is obvious that without global knowledge or mediation mechanisms, Firm2 has no way to process O1. A further examination shows the following problems:

- Two firms have different source schemas, e.g. product(fridge(clr, prc, qty)) versus catalogue(réfrigérateur(couleur, prix, quantité)).
- Firm2 cannot understand the semantic concepts of the Firm1. For instance, Firm2 cannot understand what “clr”, “prc” or “qty” refer to.
- For Firm2, the number “990” and “400” are not clear, which include the implicit meanings that only Firm1 can understand such as currency and scalar.

- Terms and values in O1 include a different natural language that is different from Firm2.

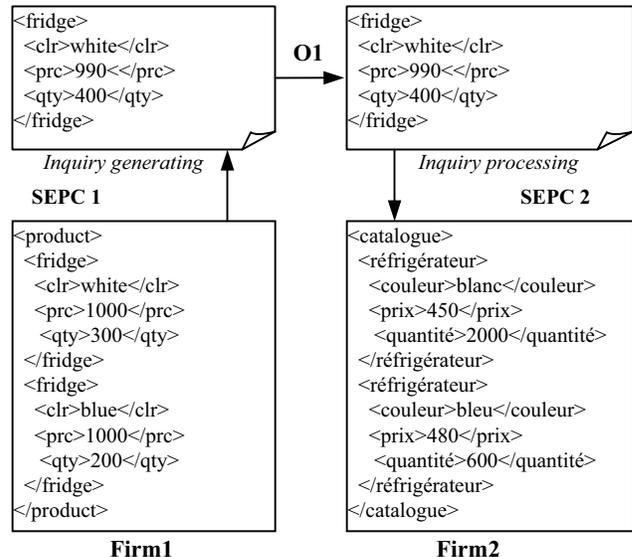


Fig. 1: A motivational example

To solve these problems, this paper proposes a novel *concept mapping approach* for transforming a set of heterogeneous product concepts from one context to another context through several mapping structures that bridge adjacent contexts.

To expand the approach discussion, this paper is organised as follows: section 2 discusses the related approaches. Section 3 formalises concept maps between two contexts, section 4 describes transformation processes through which heterogeneous product concepts are bridged. The final section provides some discussion, concludes the paper and points out the future work.

2. Related Works

Traditionally, mediation strategy is often recommended for semantically connecting heterogeneous information sources by subscribing conflict resolution services to a *mediator* (e.g. COIN mediator [7]). To facilitate this task, research has focused on the invention of inference systems (e.g. facilitator [18]), which allow heterogeneous product information exchanged between various EPCs to be intelligently compared and semantically inferred. By doing so, heterogeneous product information is appropriately transformed (e.g. NetAcademy [16]). Two distinctive mediation strategies are tight-coupling strategy and loose-coupling strategy, though their dividing line becomes unclear when layered approaches (e.g. [22], [24] and [8]) are adopted in practice.

In systems subscribing to tight-coupling approach (e.g. MEPC [17], Internet EPC [29], MEMO [27]), system administrators are responsible for detecting and resolving conflicts based on a single globally shared vocabulary (e.g. SIMS [1]), which provides for the canonical representation. A shared vocabulary is useful for mediating heterogeneous concepts. Nevertheless, if heterogeneous EPCs have different structures and semantics on the domain, the vocabulary definition commitment becomes difficult. In addition, the change of source EPCs' structures and semantics will also arise inconsistency between EPCs and the domain-wide shared vocabulary.

Loose-coupling approach, on the other hand, believes that a global vocabulary is insufficient for any non-trivial number of sources. Instead of detecting and resolving conflicts *a priori*, systems based on loose-coupling strategy (e.g. Smart/Virtual Catalogue [18], OLI [6], MOMIS [2], and MACS [19]) undertake conflict detection and resolution on product information receivers who have independent vocabulary systems that interact with a limited subset of sources each time. Based on the practices of adopted vocabulary systems, two types of mediating approaches are used: developing multiple vocabularies for multiple sources and linking them with inter-vocabulary mapping, or developing layered vocabularies that are hierarchically arranged under a global vocabulary. The former is a *multiple vocabulary approach* where each information source is described by its own vocabulary (e.g. MACS [19]). The advantage of this approach is that no common and minimal definition commitment about one global ontology/thesaurus is needed. Each source vocabulary can be developed without respect to other sources or their vocabularies. This vocabulary architecture can simplify the integration task and supports the change of sources. Nevertheless, the lack of a common vocabulary makes it difficult to compare different source ontologies/thesauri. To overcome this problem, an additional representation formalism defining the inter-vocabulary mapping is needed (e.g. RDFS mapping meta-ontology [23] and MACS linking approach [19]). The inter-vocabulary mapping identifies semantically corresponding terms of different source vocabularies. However, the mapping has to consider different structures and semantics on a domain, which makes inter-vocabulary mapping difficult to define, especially the number of participated sources increases to millions of EPCs (e.g. SMEs' ad hoc EPCs). In the context of unknown number of participating sources, mapping rules, in practice, are impossible to cover all the semantic relations between heterogeneous sources. Thus, inaccurate concept mapping becomes severe when automation programs are applied (e.g. "incompleteness" and "false values", etc. [5]). For the *layered vocabulary approach* ([22], RDFS [23], OLI [6] and Q-Calculus [8]), multiple independent vocabularies have higher-level shared vocabularies, which again have a global vocabulary. For example, in Smart Catalog and Virtual Catalog [18], local ontologies over local EPCs and global ontology are mediated through a set of translating ontologies along with a set of inference rules. The Q-Calculus [8] provides two or more layers of shared Q-Vocabularies to mediate information sources. Mediating vocabularies could resolve semantic conflicts if the local vocabularies of both senders and receivers understand the mediating vocabularies. Nevertheless, inheriting the problems of both single shared vocabulary and multiple vocabulary approaches, layered vocabulary approach has still not solved the issue of scaling up to unlimited number of ad hoc EPCs.

Mediation approach has many merits to mediate many existing heterogeneous product representations if the participated sources are limited in number and known in both structures and semantics (e.g. integrating limited number of heterogeneous product standards [23][2] or domain-wide known sources [8]).

In response to the limitations of existing mediation approaches, this paper proposes a novel *concept mapping approach* to accurately and automatically transform concept semantics from one source to another. Serving as a part of CONEX research project [9][10][11][12] [13][14], some research results of CONEX are directly used in this paper, which are:

- A *CONEX concept* is a *recursive vector concept* in the form: $c_1(c_{1,1}, \dots, c_{1,n}(\dots, (c_{1,m\dots 1}, \dots, c_{1,m\dots n}), \dots))$, which can be linearised as $(c_1, c_{1,1}, \dots, c_{1,n}, \dots, c_{1,m\dots 1}, \dots, c_{1,m\dots n})$. Its generic human readable form is $c_{1.i\dots j}$, which is gene-alike, carrying the hierarchical context information of original semantic community [9][11].
- Two heterogeneous product concepts in two different contexts can be interoperable if both can connect to a common context (but not necessarily be direct) [10][11].
- Two heterogeneous contexts cannot directly connect to each other through automation programs because of *modality judgment* issue [3]. Thus, collaboration between product concept designers is needed on *collaboration mechanism* [14]. The collaborative results are *concept maps* between two contexts, which will be formalised in Section 3.
- *CONEX framework F* for interoperation between two heterogeneous information sources is a 6-tuple $\langle S, L, C, \varphi, \Theta, \theta \rangle$ [13]. S refers to a labelled multi-set, where each set is a *source catalogue* (SEPC). L refers to a labelled multi-set, where each set is a *local catalogue* (LEPC). C refers to a labelled multi-set, where each is a *common catalogue* (CEPC). φ is a common-common context mapping structure, functioning to map concepts of two CEPCs. Θ is a local-common context map, functioning to map concepts of LEPC and CEPC. θ is a source-local map, functioning to map concepts of LEPC and SEPC.
- Product concepts travel from one context to another context along a *concept supply chain*: $SEPC1 \leftrightarrow \theta_1 \leftrightarrow LEPC1 \leftrightarrow \Theta_1 \leftrightarrow CEPC1 \leftrightarrow \varphi_2 \leftrightarrow CEPC2 \leftrightarrow \Theta_2 \leftrightarrow LEPC2 \leftrightarrow \theta_2 \leftrightarrow SEPC2$.

Employing the CONEX framework, vector concept and collaboration mechanism, the being discussed concept mapping approach will not only enable concept users to *automatically* exchange product information but also ensure *exactness* [9] of transforming concept semantics between unlimited ad hoc EPC sources. This is a key difference of our approach from existing mediation approach that can either mediate the known number of heterogeneous sources or automate the transformation of large number of sources but arise inaccuracy due to inappropriate inference mechanism in mediating systems.

3. Formalising Concept Mapping Structure

To build maps θ , Θ and φ between SEPC, LEPC and CEPC, this section investigates how to formally construct a map in which heterogeneous concepts are semantically equivalent.

For any concept in two given contexts (e.g. an SEPC, an LEPC, a CEPC or an inquiry order OI in an LEPC), the truth or falsity of a statement about the concept in one context does not

necessarily leads to the truth or falsity of a statement about the concept in another context. Whether the truth or falsity statements in both contexts could apply to a given concept is determined by human judgment of concept creators. Inspired by the work of McCarthy et al [20][21] about the assertion of a proposition in a given context (note: for McCarthy, $ist(c, p)$, where c is a context and p is a proposition), we declare that two true statements about a product concept in two contexts is formalised using the assertion:

$$\Gamma \vdash map[is(c(x_1), \sigma_1), is(c(x_2), \sigma_2)].$$

This form suggests that the statements of σ_1 and σ_2 are both true (“is”) to a concept c in two contexts x_1 and x_2 only if there exists a concept map that can make c semantically consistent to both context x_1 and x_2 . The mapping (“map”) of the concept c of x_1 and x_2 is a *human judgment declaration* to confirm that c in x_1 and x_2 is exactly the same in meaning. This judgment declaration \vdash is made by collaboration Γ between the concept creators of x_1 and x_2 .

For example, in two different electronic product catalogues EPC1 and EPC2, both may use a term “refrigerator” to refer to a certain product. Without any collaborative negotiation between EPC1 and EPC2, we have no idea what is the exact product that EPC1 or EPC2 refers to because of modality judgment issue [3]. Several situations may happen:

- (A): $is((refrigerator(EPC1), \text{“An appliance, a cabinet, or a room for storing food or other substances at a low temperature”}))$.
- (B): $is((refrigerator(EPC2), \text{“A soft isolation bag for keeping food or drink at a temperature around 3 to 8 degree centigrade”}))$.
- (C): $is((refrigerator(EPC3), \text{“domestic refrigerator”}))$.

It is obvious that (A) and (B) are different products though they are related to certain kind of cooling systems. The (A) and (C) may be the same but the scope of the category range may be different.

In another case, if EPC1 uses the term “fridge” to substitute “refrigerator”, that is, changing $c_1(x_1)$ to $c_2(x_1)$, then we have a different product concept form as follows:

- (D): $is((fridge(EPC1), \text{“An appliance, a cabinet, or a room for storing food or other substances at a low temperature”}))$.

From the statement, we know that (A) and (D) are, in fact, exactly the same.

The demonstration explains that for the form $is(c(x), \sigma)$, the statement σ is autonomous and its referred product concept c is only understandable in its own influential context x . It also indicates that how to represent a concept term such as “refrigerator”, “fridge” or “réfrigérateur” is irrelevant as long as, for two given context x_1 and x_2 , if their statements are semantically the same, then $c_1(x_1)$ and $c_2(x_2)$ are the same. To declare that they are *semantically equal* (“ \equiv ”), we have

$$\Gamma \vdash c_1(x_1) \equiv c_2(x_2).$$

It suggests that collaboration Γ between concept creators of x_1 and x_2 is necessary. To generalise, we formalise it by using the assertion of a *map*:

$$\Gamma \vdash map[is(c_1(x_1), \sigma_1), is(c_2(x_2), \sigma_2)].$$

where c_1 and c_2 can be any symbols to notate corresponding contextual statement σ_1 and σ_2 , which in our CONEX

project are concept *annotations* in a *denotative concept structure* [11][13].

The *map* above refers to the ubiquitous semantic connection between heterogeneous EPCs, in which different contextual concepts are integrated. The Γ , as a collaboration mechanism, assists concept creators to make *concept maps* between contexts. For example,

$$\Gamma \vdash map[is(cream(\infty \text{ colour } \infty \text{ refrigerator } \infty \text{ EPC1}), \text{“cream is colour of domestic refrigerator”}), is(\text{奶白色 } (\infty \text{ 颜色 } \infty \text{ 电冰箱 } \infty \text{ EPC2}), \text{“奶白色是电冰箱的颜色”})].$$

In this form, “ ∞ ” specifies the detailed hierarchical context with “ $c_1 \infty c_2$ ” to refer to “ c_1 is in the context of c_2 ”. This new form exactly depicts where a concept should be for concept interpreters in a concept hierarchy $c_1(c_{1.1}(\dots(c_{1.1.m}(\dots(c_{1.1.m.n}), \dots), \dots), \dots), \dots)$. The generalisation of the above form is:

$$\Gamma \vdash map[is(v_1(c_1 \infty \dots \infty c_n \infty x_1), \sigma_1), is(v_2(c'_1 \infty \dots \infty c'_m \infty x_2), \sigma_2)].$$

By this generic form, the ambiguous mapping of *constant values* between two contexts could be reduced to minimum and accuracy is maintained. For example, “cream” will not be translated into “奶油” (cream for food in Chinese) or “冰激凌” (ice cream in Chinese).

4. Transforming Product Concepts on CONEX

Based on the formalised concept map between two contexts, this section describes a heterogeneous concept transformation algorithm between contexts on the CONEX framework for effective product concept exchange, which includes five sub-algorithms: source-to-local context transformation (SLCT), local-to-common context transformation (LCCT), common-to-common context transformation (CCCT), common-to-local context transformation (CLCT) and local-to-source transformation (LSCT).

4.1 Active Context Transformation

We assume that the CONEX systems have already established concept structures of CEPCs, LEPCs and SEPCs in XPM formats [11] and filled them with necessary concepts: common concepts for CEPCs, local concepts for LEPCs, source concepts for SEPCs (legacy data that originally exist). With this assumption, we provide the following definitions:

Definition 1: Active Context (“actCtx”)

An *active context* actCtx is a set of concepts c_i created on a sender’s context S and related to a receiver’s context R . It is formulated based on S and expected to be understood by R . Strictly, if R understands S , then

$$\forall c_i \in actCtx \bullet actCtx \subseteq (S \wedge R) \quad \square$$

By this definition, SEPC, LEPC and CEPC are all contexts and the inquiry O1 is an active context, which is created on context SEPC1 and expects to be understood by adjacent context LEPC1 for creating new active context ELO (see Fig. 2). The similar process continues until an *actCtx* TO1 is created (see Fig. 2). However, as we have known, the contexts SEPC, LEPC and CEPC are semantically independent (“||”) such that:

$$SEPC1 \parallel LEPC1 \parallel CEPC1 \parallel CEPC2 \parallel LEPC2 \parallel SEPC2.$$

To have them understood with each other for transforming O1 to TO1, maps between two adjacent contexts are inserted through collaboration Γ . The map (written in “ Ξ ”) as formalised in Section 3 is represented in the form:

$$\Xi(ctx_a, ctx_b),$$

where ctx_a and ctx_b represent two adjacent EPCs.

Since an $actCtx$ is generated on ctx_a and understood by ctx_b , we have the form:

$$actCtx \subseteq \Xi(ctx_a, ctx_b).$$

Thus, for any $actCtx_a$, if $actCtx_a \subseteq ctx_a$ and $actCtx_a \subseteq \Xi$, then we can always find an $actCtx_b \subseteq ctx_b$ and $actCtx_b \subseteq \Xi$ such that $actCtx_a \rightarrow actCtx_b$. This suggests a feasible active context transformation scenario such that:

$$\begin{aligned} O1(SEPC1) &\leftrightarrow \Xi(SEPC1, LEPC1) \leftrightarrow ELO(LEPC1) \leftrightarrow \\ &\Xi(LEPC1, CEPC1) \leftrightarrow ECO(CEPC1) \leftrightarrow \Xi(CEPC1, CEPC2) \\ &\leftrightarrow FCO(CEPC2) \leftrightarrow \Xi(CEPC2, LEPC2) \leftrightarrow FLO(LEPC2) \leftrightarrow \\ &\Xi(LEPC2, SEPC2) \leftrightarrow TO1(SEPC2). \end{aligned}$$

The successful realisation of this scenario will accurately transform inquiry O1 of SEPC1 to TO1 that SEPC2 understands.

4.2 Concept Mapping via Concept Identifiers

The above transformation scenario is workable because an EPC as a context is a set of concepts. Within each concept = (iid , annotation, link, options), $iid \leftarrow annotation (An) \leftarrow$ (link, options). Thus, an iid has represented the full semantics of a concept and is qualified to replace a complex concept. Applying for $iids$, we can state that two adjacent EPCs understand with each other if and only if their sets of $iids$ are aligned in a set of maps, formally:

$$EPC_a \equiv EPC_b \text{ IFF } \forall iid_i \in EPC_a, iid_j \in EPC_b \bullet \Xi(iid_i, iid_j).$$

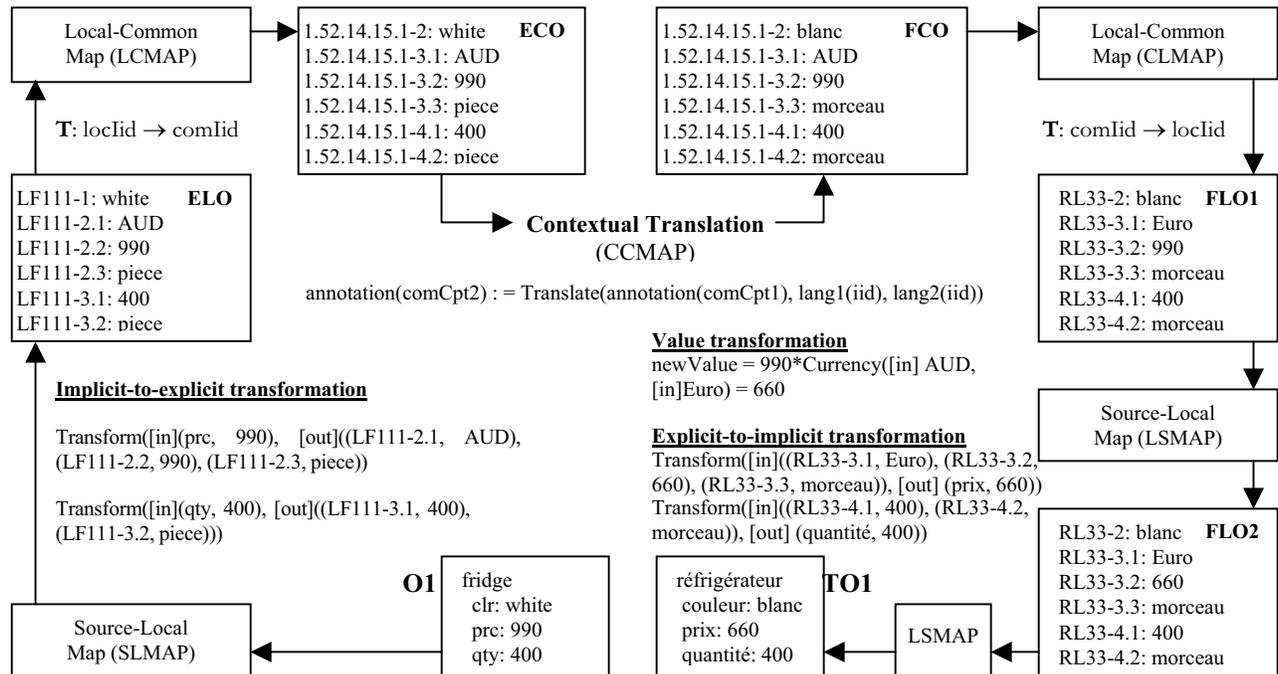


Fig. 2: Concept transformation between contexts by example

By this formalisation, we achieve the following maps along concept supply chain:

Definition 2: IID-based Concept Maps

(1) Common-common map (CCMAP)

$$CCMAP: \Xi(CEPC_a, CEPC_b)$$

$$\text{IFF } \forall comId_i \in CEPC_a, CEPC_b,$$

$$comAn_i \in CEPC_a, comAn'_i \in CEPC_b \bullet$$

$$comId_i \leftarrow \Xi(comAn_i, comAn'_i)$$

where $comId$ is common concept identifier and $comAn$ is common annotation for the concept meaning.

(2) Local-common map (LCMAP)

$$LCMAP: \Xi(LEPC, SubC \subseteq CEPC)$$

$$\text{IFF } \forall locId_i \in LEPC, \forall comId_i \in SubC \bullet \Xi(locId_i, comId_i)$$

where $locId$ is local concept identifier.

(3) Local-source map (LSMAP)

$$LSMAP: \Xi(LEPC, SEPC)$$

$$\text{IFF } \forall locId_i \in LEPC, \forall srcId_i \in SEPC \bullet$$

$$\Xi(locId_i, srcId_i) \leftarrow \Xi(locAn_i, srcAn_i)$$

where $srcId$ is source concept identifier, $srcAn$ refers to a unique term in source schema (e.g. “fridge”, “clr” or “prc” in the schema of SEPC1), and $locAn$ refers to the annotation such that $locId \leftarrow locAn$. \square

In CONEX approach, source concept identifier is a path expression corresponding to a term in an SEPC schema. For example, for relational table “dbName, dataTable (name, color, price, quantity): (fridge, white, 1000, 300)”, the $srcId$ of name is: $dbName.dataTable.name$. Specifically, in CONEX approach, the source iid in relational database is a path expression of LOREL [26] in the form of “A.B.C.D”, and the $srcId$ in XML document is an XPath expression [4].

Using $iids$ to represent interoperable concept semantics provides us an opportunity to build simple concept map between two contexts in the form:

$$\exists (iid_1, iid_2).$$

This form allows us to rewrite *concept supply chain* for active context transformation shown as in Fig. 2 in an unambiguous way:

- $O1 \rightarrow \sum \exists (srcLid_i, locLid_i) \rightarrow ELO$
- $ELO \rightarrow \sum \exists (locLid_i, comLid_i) \rightarrow ECO$
- $ECO \rightarrow \sum comLid_i \rightarrow FCO$
- $FCO \rightarrow \sum \exists (locLid_i, comLid_i) \rightarrow FLO$
- $FLO \rightarrow \sum \exists (srcLid_i, locLid_i) \rightarrow TO1$

Thus, the active context transformation includes five major steps. The complete *iid*-based context transformation process is diagrammed as an example in Fig. 2.

4.3 Source-Local Context Transformation

The *source-local context transformation* (SLCT) is to transform the active context O1 into an active context ELO. Key to this transformation is to transform a set of possibly implicit and reified source concepts into a set of explicit and reified local concepts. Specifically, it is to transform a set of reified *srcLids* into a set of reified *locLids*. A *reified iid* refers to a form of (*iid*, *value*).

Given a source-local mapping context SLMAP, O1 can be transformed into ELO if the following conditions are satisfied:

- (1) $\forall ctxAn_i \in O1, \exists ! srcAn_i, srcLid_i, locLid_i \in SLMAP \bullet$
 $(ctxAn_i \rightarrow srcAn_i) \wedge (srcAn_i \rightarrow srcLid_i) \wedge \exists (srcLid_i, locLid_i)$
- (2) $\forall ctxAn_i \rightarrow Implicit(srcAn_i) \rightarrow srcLid_i \bullet$
 $srcLid_i \rightarrow locLid_i \{locLid_{i,1}, \dots, locLid_{i,n}\}$
- (3) $\forall locLid_i \{locLid_{i,1}, \dots, locLid_{i,n}\},$
 $\exists ! locLid_{i,k} \rightarrow Reified(Value \leftarrow srcLid_i \leftarrow Implicit(srcAn_i))$
- (4) $\forall locLid_i \{locLid_{i,1}, \dots, locLid_{i,n}\} \bullet$
 $locLid_{i,1} \rightarrow defVal_{i,1}, \dots, locLid_{i,n} \rightarrow defVal_{i,n} \wedge NOT(locLid_{i,k})$

The first condition suggests that each concept (*ctxAn*) of active context O1 must fall in the map of SLMAP and correspond to a source concept *srcAn* from the source schema of SEPC1. This *srcAn* has an *srcLid* that is mapped onto a *locLid* of O1, which are both in SLMAP.

The second condition suggests that if a concept *ctxAn* of O1 corresponds an implicit concept *srcAn* of SEPC1 in SLMAP, then the corresponding *locLid* of *srcLid* determined by *srcAn* must be expanded into a set of explicit *locLids* in terms of a small two level concept tree.

The third condition suggests that, for expanded concept tree *locLid*, one of leaf *locLid* carries a value transferred from the value of the implicit concept of SEPC.

The fourth condition suggests that, for each expanded explicit leaf *locLids* except for the one that has value transferred, it is assigned a default value.

Typically, the result of ELO transformed from O1 can be written in an XPM document [11] of the following:

```
<context lang="English" url="www.lepc1.com/inquiry">
  <concept locLid = "LF111-1">white</concept>
  <concept locLid = "LF111-2.1">AUD</concept>
  <concept locLid = "LF111-2.2">990</concept>
  <concept locLid = "LF111-2.3">piece</concept>
  <concept locLid = "LF111-3.1">400</concept>
  <concept locLid = "LF111-3.2">piece</concept>
</context>
```

This sample result is an active context that includes a *context root* "context" whose attributes are sender's information for remote processing. The body of the context consists of a set of exchanging concepts - *concept leaves*, whose parents that are missing can be inferred through their *locLids*, for instance, the described product is "LF111" that refers to "fridge". The omission of intermediate concept nodes is a peculiar feature of CONEX system, which provides *dynamic node calculation* by IID.

For better understanding, we clarify two points. (1) The input concepts such as "fridge", "clr", "prc" and "qty" are exactly semantically consistent with the *annotation* of concept definition (*iid*, *annotation*) in LEPC1. This is not an assumption but the result from the collaboration mechanism " $\Gamma \vdash \Xi$ ". (2) The local identifiers such as "LF111, LF111-2, ..." are naturally classified as a computable tree where their traceable roots are products. For example, "LF111" is a root, and "LF111-1, LF111-2" are children of "LF111". Similarly, "LF111-2.1, LF111-2.2, LF111.2.3" are the children of "LF111-2". The example is simplified but it is strictly consistent with the concept hierarchy mentioned in Section 2.

4.4 Local-Common Context Transformation

The *local-common context transformation* (LCCT) is to transform the active context ELO into ECO. Core to this transformation is to compare whether concepts of ELO is in a common context CEPC1. If so, then ELO can be transformed into ECO.

Given a local-common mapping document LCMAP, ELO is transformed to ECO if:

- (1) $\forall locLid_i \in ELO, locLid_j \in LCMAP \bullet \exists (locLid_i, comLid_j)$
- (2) $\forall locLid_i \rightarrow val_i \in ELO, val_j \leftarrow comLid_j$

The first condition suggests that all *locLids* in ELO must fall in LCMAP, where *locLids* and *comLids* are mapped. The Second condition suggests that the values carried by *locLids* must be transferred as the reified values of *comLids* accordingly.

The transformation result ECO in XPM form is illustrated in the following:

```
<context lang="English" url="www.lepc1.com/inquiry">
  <concept comLid = "1.52.14.15.1-2">white</concept>
  <concept comLid = "1.52.14.15.1-3.1">AUD</concept>
  <concept comLid = "1.52.14.15.1-3.2">990</concept>
  <concept comLid = "1.52.14.15.1-3.3">piece</concept>
  <concept comLid = "1.52.14.15.1-4.1">400</concept>
  <concept comLid = "1.52.14.15.1-4.2">piece</concept>
</context>
```

Exceptions may happen in local-common transformation if local concepts are not found in LCMAP. If such cases happen, a *localisation* process [14] is required to generate local-common concept maps in LCMAP. If no common concepts are found for localisation, a *globalisation* process should be run [14] for collaborative design of common concepts between the designers of CEPC1 and LEPC1

4.5 Common-Common Context Transformation

The *common-common context transformation* (CCCT) is to transform ECO of CEPC1 into FCO of CEPC2. The task of this transformation is to examine the common concept consistency

between CEPCs and to translate reified values from one language to another language.

Given that CEPC2 has context-aware language translation programs, ECO can be transformed into FCO if the following conditions are satisfied:

- (1) $\forall comLid_i \in ECO, comLid_i = comLid_j \in CEPC2$
- (2) $\forall comLid_i \rightarrow comVal_i \in ECO,$
 $comVal_j \leftarrow Translate(comVal_i, lang(ECO), lang(CEPC2)),$
 $comVal_j \leftarrow comLid_j \in FCO$

The first condition suggests that the *comLids* of ECO must fall in CEPC2. The second condition suggests that the reified values of active context ECO be translated into the language of CEPC2 through context-aware translation programs. The expected transformed result is:

```
<context lang="French" url="www.lepc1.com/inquiry">
  <concept comLid = "1.52.14.15.1-2">blanc</concept>
  <concept comLid = "1.52.14.15.1-3.1">AUD</concept>
  <concept comLid = "1.52.14.15.1-3.2">990</concept>
  <concept comLid = "1.52.14.15.1-3.3">morceau</concept>
  <concept comLid = "1.52.14.15.1-4.1">400</concept>
  <concept comLid = "1.52.14.15.1-4.2">morceau</concept>
</context>
```

A possible issue may arise in *reified value translation*, that is, the translation program cannot accurately translate the values. For example, a reified value “cream” may be wrongly translated into “奶油” (a kind food in Chinese) or “冰激凌” (ice cream in Chinese). CONEX concept mapping approach solves this problem by providing correct context information to translation program. In our approach, a common concept implies a hierarchical context. For example, the common concept *iid* 1.52.14.15.1-2 implies a specific context of colour of refrigerator of domestic appliances.

```
iid(1.52.14.15.1-2) :=
  1:ConexEPC
  52:domestic Appliances and Supplies and Consumer
  Electronic Products
  14:domestic appliances
  15:domestic kitchen appliances
  1:domestic refrigerators
  2:colour
```

This gene-like *iid* of meta-concept “colour” ensures that translation programs can receive correct translation context for translating reified values. Under this condition, “cream” will be correctly translated into 奶白色 (“cream colour”). Providing *specific term translation context* is a special feature of CONEX concept exchange, because each *iid* implies a classification context of concept from a wide category to a specific context.

4.6 Common-Local Context Transformation

The *common-local context transformation* (CLCT) is to transform FCO to FLO1 (see Fig. 2) of LEPC2. Given a common-local mapping document CLMAP, FCO can be transformed into FLO1 if the following conditions are satisfied:

- (1) $\forall comLid_i \in FCO, comLid_i \in CLMAP \bullet \exists (comLid_p, locLid_i)$
- (2) $\forall comLid_i \rightarrow val_i \in FCO, val_i \leftarrow locLid_i$

These two conditions guarantee that FCO is transformed to FLO1 in following form:

```
<context lang="French" url="www.lepc1.com/inquiry">
  <concept locLid = "RL33-2">blanc</concept>
  <concept locLid = "RL33-3.1">AUD</concept>
  <concept locLid = "RL33-3.2">990</concept>
  <concept locLid = "RL33-3.3">morceau</concept>
  <concept locLid = "RL33-4.1">400</concept>
  <concept locLid = "RL33-4.2">morceau</concept>
</context>
```

By this transformation, common concepts have transformed to local concepts.

4.7 Local-Source Context Transformation

The *local-source context transformation* (LSCT) is to transform FLO1 to TO1 so that legacy systems can understand the incoming inquiry. Two tasks involved in this transformation are: (1) to transform incoming reified values of FLO1 to FLO2 (see Fig. 2) if incoming unit types are the same but their values are different from the default values of LEPC2; and (2) to transform explicit concepts of FLO2 to implicit concepts of TO1 that is semantically consistent with the source LEPC2.

For the first task, two conditions must be satisfied:

- (1) $\forall locLid_i \rightarrow locVal_i \in FLO1,$
 $locLid_i \in \Xi_i(locLid_i, transformFunction) \in LSMAP$
- (2) $locLid_i$ has parent concept identifier $locLid_p,$
 Case 1: for $\Xi_i,$ transformFunction = UnitTransform
 and $(defVal_i \leftarrow locLid_i) \in LEPC2$
 UnitTransform($[in] locVal_i, [in] defVal_i, [out] valFactor_i$)
 Case 2: for $\Xi_i,$ transformFunction = ValTransform
 $locLid_i \rightarrow locVal_i := ValTransform([in] comVal_i,$
 $[in] valFactor[k])$

The first condition suggests *locLids* of FCO1 that has reified value $locVal_i$ must fall in a *concept map* including a transformation function belonging to LSMAP. The second condition suggests that if the reified value of *locLid* is different from the default value of LEPC2, then a unit transformation function is triggered to calculate the modifying value *valFactor*. If a value function is triggered, then the reified value of *locLid* must be calculated by multiplying the *valFactors*. For example,

- A *valFactor* is generated based on the existing difference of *locVal* and *defVal* (e.g. UnitTransform (AUD, Euro) = 0.6667, supposing 1 Euro=1.5 AUD).
- An adjusted value is calculated by multiplying a set of *valFactors* (e.g. ValTransform(990, 0.6667)=660).

The result of FLO2 transformed from FLO1 is presented in the following:

```
<context lang="French" url="www.lepc1.com/inquiry">
  <concept locLid = "RL33-2">blanc</concept>
  <concept locLid = "RL33-3.1">Euro</concept>
  <concept locLid = "RL33-3.2">660</concept>
  <concept locLid = "RL33-3.3">morceau</concept>
  <concept locLid = "RL33-4.1">400</concept>
  <concept locLid = "RL33-4.2">morceau</concept>
</context>
```

Transformation functions of *UnitTransform* and *ValTransform* are imported from CEPC2 in the stage of LSMAP design, i.e., at the time of local concept design.

For the second task, given local-source mapping document LSMAP, FLO2 can be transformed into TO1 if the following conditions are satisfied:

- (1) $\forall locLid_i \in FLO2, locLid_i \in LSMAP \wedge \exists_i (locLid_i, srcLid_i, conceptType) \bullet srcLid_i \in SEPC2$
- (2) $\forall locLid_i \rightarrow locVal_i \in FLO2,$
 - Case 1: for $\exists_i, conceptType = \text{“atomicConstantType”}$
Expand $srcLid_i$ into a tree such that
 $c_1(\dots, c_{1.i}(\dots, (c_{1.i\dots i}), \dots))$
 $srcAn_i := c_{1.i\dots i}$
 $reifiedValue(c_{1.i\dots i}) := locVal_i$
 - Case 2: for $\exists_i, conceptType = \text{“atomicUnitType”}$
Ignore $locLid_i$ and $locVal_i$
 - Case 3: for $\exists_i, conceptType = \text{“atomicValueType”}$
Find Parent($srcLid_i$)
Expand Parent($srcLid_i$) into a tree
 $c_1(\dots, c_{1.i}(\dots, (c_{1.k\dots k}), \dots))$
Merge the trees if previous trees exist
 $srcAn_i := c_{1.k\dots k}$
 $reifiedValue(c_{1.k\dots k}) := locVal_i$

The first condition suggests that *locLids* of FLO2 must fall in LSMAP, where they are mapped onto *srcLids* of SEPC2. The second condition suggests that if the mapped *srcLid* is an explicit constant concept, then the *locVal* is directly assigned to the leaf of *srcLid*. If it is constrained to implicit default value, then the default value is ignored in the context of SEPC systems. If it is constrained to a value but brought by an explicit *locLid*, then the parent of the implicit *srcLid* takes over the value. This is, in fact, a process of value restoration, which is a reverse of source-local transformation. By satisfying these conditions, FLO2 is transformed into TO1, which is understandable to SEPC2:

```
<catalogue>
  <réfrigérateur>
    <couleur>blanc</couleur>
    <prix>660</prix>
    <quantité>400</quantité>
  </réfrigérateur>
</catalogue>
```

4.8 Concept Transformation Algorithm

With the above computing of concept transformation between contexts, this part generalises the process into an algorithm, called *heterogeneous concept transformation algorithm* (HCT), which includes five sub-steps SLCT, LCCT, CCCT, CLCT and LSCT.

Pre1: CEPC1, ..., CEPCn

Pre2: SLMAP, LCMAP, CLMAP and LSMAP

Pre3: Translate([in] val, [in] lang1, [in] lang2) in CEPCs

Input: ctxDocIn in SEPC1 domain

Output: ctxDocOut in SEPC2 domain

HCT(ctxDocIn)

```
IF lid(ctxDocIn)  $\subseteq$  SLMAP THEN // SLCT Step
  MakeExplicit ctxDocIn
  Swap iids
  ctxDocLA  $\leftarrow$  ctxDocIn
IF lid(ctxDocLA)  $\subseteq$  LCMAP THEN // LCCT Step
  Swap iids
  ctxDocCA  $\leftarrow$  ctxDocLA
```

```
IF Sender(ctxDocCA)  $\subseteq$  CEPCa AND // CCCT Step
  Receiver(ctxDocCA)  $\subseteq$  CEPCb THEN
  FOR iidi, vala, valb  $\leftarrow$  iidi
    valb := Translate(vala, langa, langb)
  ctxDocCB  $\leftarrow$  ctxDocCA
IF iid(ctxDocCB)  $\subseteq$  CLMAP THEN // CLCT Step
  Swap iids
  CtxDocLB  $\leftarrow$  ctxDocCB
IF iid(ctxDocLB)  $\subseteq$  LSMAP THEN // LSCT1 Step
  Iterate {
    Iterate UnitTransform
    ValTransform}
  ctxDocLC  $\leftarrow$  ctxDocLB
IF iid(ctxDocLC)  $\subseteq$  LSMAP THEN // LSCT2 Step
  Swap iids
  MakeImplicit ctxDocLC
  ctxDocOut  $\leftarrow$  ctxDocLC
```

The HCT algorithm ensures heterogeneous concepts are transformed between contexts. By this algorithm, heterogeneous contextual documents such as the inquiry “O1” and TO1 can understand with each other.

5. Conclusion

This paper has proposed a novel heterogeneous concept mapping approach to solve semantic interoperability issue. The solution is based on CONEX framework [13], where heterogeneous product information is stored in source catalogues (SEPCs), local catalogues (LEPCs) and common catalogues (CEPCs). As a contribution of this paper, *maps* between ad hoc EPCs are investigated to map heterogeneous product information of adjacent catalogues. By these maps, a *heterogeneous concept transformation* algorithm is designed for transforming heterogeneous concepts along concept supply chain: source-local mapping, local-common mapping, common-common-mapping, common-local mapping and local-source mapping.

The evaluation of this algorithm is mainly against the *exactness* [9]: whether a piece of semantic product information can be faithfully conveyed from the sender to receiver. Since meta-concepts (e.g. colour and couleur) are collaboratively mapped [14] based on *financial cost minimisation criterion* [15], they are semantically precise between all participated parties. The only possible semantic inconsistency between sender and receiver is reified concept values (e.g. “cream” value for “colour” concept). CONEX solves this problem by providing run-time translation context of Section 4.5 and expects context-aware translation program can incorporate into this solution. The correctness of this algorithm has been proved in the CONEX prototype demonstrated as a Transformer in CONEX project website.

In summary, this paper has contributed a heterogeneous concept mapping approach, which bridges various contexts of different semantic communities. By this approach, any contextual documents could transparently interoperate with each other disregarding their natural languages, data structures, semantic encoding and context referencing.

A future work of this paper is to refine heterogeneous concept transformation algorithm. We are planning to facilitate an experiment on this algorithm to observe its accuracy on concept mapping, especially on value translation by plugged-in translation programs.

6. Acknowledgement

We thank four anonymous reviewers for their insight comments, which were invaluable for improving the presentation of this work.

Reference

- [1] Arens, Y., Hsu, C. and C. Knoblock, "Query Processing in the SIMS Information Mediator", in: *Advanced Planning Technology*, AAAI Press, California, USA, 1996.
- [2] Bergamaschi, S. Guerra, F. and M. Vincini, "A Data Integration Framework for e-Commerce Product Classification", in: I. Horrocks and J. Hendler (Eds.): *ISWC 2002*, LNCS 2342, 2002, pp. 379-393.
- [3] Chandler, D., *Semiotics for Beginners*, <http://www.aber.ac.uk/media/documents/S4B/semiotic.html>.
- [4] Clark, J. and S. DeRose, "XML Path Language (XPath) Version 1.0, *W3C Recommendation* 16 November 1999, <http://www.w3.org/TR/xpath>.
- [5] 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*, Vol. 16, No. 4, 2001 pp. 54-59.
- [6] Fensel, D., Harmelen, F., Horrocks, I., McGuinness, D. and P. Patel-Schneider, "OIL: An Ontology Infrastructure for the Semantic Web", *IEEE Intelligent Systems*, March/April, 2001, pp.38-45.
- [7] Goh, C., Madnick, S. and M. Siegel, Context Interchange: Overcoming the Challenges of Large-Scale Interoperable Database Systems in a Dynamic Environment, in: *ACM CIKM'94*, Gaithersburg, MD, USA, 1994, pp. 337-346.
- [8] Greunz, M. and J. Peer, "Prototype Implementation of the Q Calculus", *Universität St. Gallen*, Switzerland, 2002, <http://www.mcm.unisg.ch>.
- [9] Guo, J. and C. Sun, "Concept Exchange: Constructing Interoperable Electronic Product Catalogues in an Emergent Environment", in: *Proc. of IEEE Int'l Conf. on E-Commerce (CEC'03)*, IEEE Computer Society Press, 2003, pp. 165-172.
- [10] Guo, J. and C. Sun, "Context Representation of Product Data", *ACM SIGEcom Exchanges*, Vol. 4, No. 1, ACM Press, 2003, pp. 20-28.
- [11] Guo, J. and C. Sun, "Context Representation, Transformation and Comparison for Ad Hoc Product Data Exchange", in: *Proc. of the 2003 ACM Symposium on Document Engineering*, ACM Press, 2003, pp. 121-130.
- [12] Guo, J. and C. Sun, "Global Electronic Markets and Global Traditional Markets", *Electronic Markets*, Vol. 14, No. 1, Routledge, 2004, pp. 4-12.
- [13] Guo, J., Sun, C. and D. Chen, "Deconstruction and Reconstruction of Heterogeneous Electronic Product Catalogues for Semantic Interoperation", in: *Proc. of IEEE Int'l Conf. on E-Commerce Technology (CEC'04)*, IEEE Computer Society Press, 2004, pp 333-336.
- [14] Guo, J., Sun, C. and D. Chen, "Collaborative Integration of Product Data: Analysis, Strategy and Protocol", *Int'l Conf. on Electronic Commerce and Research (ICECR07)*, Dallas, TX, USA, June 10-13, 2004 (Accepted).
- [15] Guo, J., Sun, C. and D. Chen, "Articulating Autonomously Distributed Electronic Product Catalogues for Constructing Dynamic CONEX Net", *IEEE Conference on E-Commerce Technology for Dynamic Business*, Beijing, China, September 13-15, 2004.
- [16] Handschuh, S., Lechner, U., Lincke, D., Schmid, B., Schubert, P., Selz, D., K. Stanoevska-Slabeva, "The NetAcademy – A New Concept for Online Publishing and Knowledge Management", in: *Proc. of the 31st HICSS Conf.*, IEEE Computer Society Press, 1998.
- [17] Handschuh, S., Schmid, B. F. and Stanoevska-Slabeva, K. "The Concept of a Mediating Electronic Product Catalog", *Electronic Markets*, Vol.7, No. 3, 1997, pp. 32-35
- [18] Keller, A. M., "Multivendor Catalogues: Smart Catalogues and Virtual Catalogues", *EDI Forum: Journal of Electronic Commerce* 9(3), 1996, pp. 87-93.
- [19] Landry, P., "Multilingual Subject Access: The Linking Approach of MACS", *Cataloging & Classification Quarterly*, Vol. 37, No. 3/4, Haworth Information Press, 2004, pp. 177-191.
- [20] McCarthy, J., "A Logical AI Approach", *Computer Science Department, Stanford University*, February 6, 1996, <http://www-formal.stanford.edu/jmc>.
- [21] McCarthy, J. and S. Buvac, "Formalizing Context (Expanded Notes)", *Computing Natural Language*, A. Aliseda, R. Glabbeek and D. Westerstahl (eds), Stanford University, 1997, <http://www-formal.stanford.edu>.
- [22] Melnik, S. and S. Decker, "A Layered Approach to Information Modelling and Interoperability on the Web", in: *Proc. of the Workshop on the Semantic Web at 4th European Conf. on Research and Advanced Technology for Digital Library*, Lisbon, Portugal, September 21, 2000.
- [23] Omelayenko, B., "RDFT: A Mapping Meta-Ontology for Business Integration", in: *Proc. of the Workshop on Knowledge Transformation for the Semantic Web at the 15th European Conf. on Artificial Intelligence*, Lyon, France, 2002, pp.77-84.
- [24] 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 E-Commerce and Web Tech*, LNCS 2115, 2001, 226-239.
- [25] Phillips, C. and M. Meeker (chief authors) "*Collaborative Commerce, The B2B Internet Report*", April, Morgan Stanley Dean Witter, 2000.
- [26] Quass, D., Rajaraman, A., Sagiv, Y., Ullman, J. and J. Widom, "Querying Semistructured Heterogeneous Information", in: *Proc. of DOOD Conference*, 1995.
- [27] Quix, C., Schoop, M. and M. Jeusfeld, "Business Data Management for Business-to-Business Electronic Commerce", *ACM SIGMOD Record*, 31(1), 2002, pp. 49-54.
- [28] Robinson, M. and L. Bannon, Questioning Representations, *ECSCW'91*, Amsterdam, Holland, September 1991.
- [29] Stanoevska-Slabeva, K. and Schmid, B., "Internet Electronic Product Catalogs: An Approach beyond Simple Keywords and Multimedia", *Computer Networks* 32, 2000, pp. 701-715.