Fault-tolerant Semantic Mappings Among Heterogeneous and Distributed Local Ontologies

Abdul-Rahman Mawlood-Yunis
School of Computer Science, Carleton University
1125 Colonel By Drive
Ottawa, Ontario, K1S 5B6, Canada
armyunis@scs.carleton.ca

ABSTRACT

Overcoming semantic mapping faults, i.e. semantic incompatibility, is a vital issue for the success of semantic-based peer-to-peer systems. There are various research efforts which address the classification and the resolution of the semantic mapping fault problem, i.e. translation errors. All of the precedent research related to semantic mapping faults demonstrates one significant shortcoming. This flaw is the inability to discriminate between non-permanent and permanent semantic mapping faults, i.e. how long do semantic incompatibilities stay effective and are the semantic incompatibilities permanent or temporary? The current research examines the destructive effect of semantic mapping faults on the Emerging Semantics, i.e. bottom-up construction of ontology and proposes a solution to detect temporal semantic mapping faults. The current research also demonstrates that fault-tolerant semantic mapping will result in Emerging Semantics which are more complete and agreeable than those domain ontologies that are built without consideration for fault-tolerant semantic mapping.

Categories and Subject Descriptors

H.3.4 [**Information Systems**]: INFORMATION STORAGE AND RETRIEVAL; Systems and Software[Information networks]

General Terms

Algorithms, Design, Reliability

Keywords

Ontology Mapping, Emergent Semantics, Consensus Reaching, Heterogeneous Information sources, Temporal Faults, Interoperability, Semantic Conflicts

1. INTRODUCTION

Overcoming semantic mapping faults is a critical issue for success of the semantic based peer-to-peer (SP2P) systems. By SP2P systems we refer to P2P systems in which peers have different schemes and local mapping between schemes are used to further distribute queries among the peers. Examples of such systems include P2P knowledge management systems, P2P Semantic Web, semantic-based Web Services, and any P2P systems which require cooperation among distributed and autonomous peers with heterogeneous information sources (local ontologies).

Various precedent research efforts had the objective of the classification and the resolution of the semantic incompatibility problem[1] [6] [7] [8] [11] [12] [15] [21] [23] [27] [28]. The existing research related to semantic fault suggests that the conflicts are mainly due to the heterogeneous representation of information created by autonomous and independent information source providers. This vision has a significant shortcoming; it focuses on the information representation aspect and fails to recognize the temporal dimension of the semantic mapping fault. By the temporal dimension, we refer to the *duration* in which a mapping fault stays effective, i.e. is the fault temporary or permanent?

In the current research, we propose a simple solution to detect mapping faults. The solution is based on the *time redundancy* technique [3][4][10][25]. Time redundancy refers to the replication of the query and the verification for the query answer consistency. We choose the emergent shared semantics among distributed heterogeneous local ontologies for studying the effect of the semantic mapping faults. This is because the emergent semantics process depends entirely on the correctness of the local semantic mappings. More details about emerging shared semantics, bottom-up construction of ontology, are provided in section 4.

Augmenting our proposed solution to the emergent semantics method, we envision to build domain ontologies and generate emerging semantics which are more complete and agreeable than those domain ontologies and emerging semantics that were built without fault-tolerant semantic mappings. We will demonstrate this improvement through a detailed example.

The reminder of this paper is organized as follows: In Section 2, our previous informal presentation of the definition of fault and fault types is revised and presented more formally. In Section 3, transient and intermittent mapping faults arising from ontology modification are discussed. In Section 4, we look at the effect of semantic mapping faults on the automatic emergent shared semantics method. In Section 5,

Permission to make digital or hard copies of all or part of this work for personal or classroom use is granted without fee provided that copies are not made or distributed for profit or commercial advantage and that copies bear this notice and the full citation on the first page. To copy otherwise, to republish, to post on servers or to redistribute to lists, requires prior specific permission and/or a fee.
ONISW'08, October 30, 2008, Napa Valley, California, USA.
Copyright 2008 ACM 978-1-60558-255-9/08/10 ...$5.00.

[1]Please note that we use semantic incompatibility and semantic mapping faults interchangeably throughout this work.

a solution to the problem is proposed. In Section 6, an illustrative example is presented. In Section 7, the paper is concluded and future research directions are identified.

2. MAPPING FAULT AND TYPES

In this section we define what we mean by a semantic mapping fault, and identify different types of faults based on notions from fault tolerance literature. These are revised definitions to those presented informally in [18].

DEFINITION 1. *A* **mapping fault** *is an incorrect semantic mapping, or the failure to map between concepts from different ontologies. We say that a fault occurs when (i) a concept in one ontology is mapped to a semantically unrelated concept in a different ontology, or (ii) a concept in one ontology cannot be mapped to an existing semantically related concept in another ontology.*

Formally we can express this definition as follows. Assume we have two ontologies $\vartheta 1 = \{C, P, R\}$ and $\vartheta 2 = \{\bar{C}, \bar{P}, \bar{R}\}$, where C and $\bar{C}$ are sets of concepts, P and $\bar{P}$ are sets of concept properties, and R and $\bar{R}$ are sets of relations between concepts [13][14].

Given two semantically equivalent concepts $c \in C$ and $\bar{c} \in \bar{C}$, $c \equiv \bar{c}$, we can say that a fault occurs if either one of the following is true:

- c is mapped to a semantically unrelated concept $x \in \bar{C}$, $x \not\equiv \bar{c}$;
- c cannot be mapped to a semantically related concept $\bar{c} \in \bar{C}$, i.e. the mapping process incorrectly leads to nil.

The fault-tolerance discipline classifies faults based on their duration [25]. Accordingly, we distinguish between permanent, transient and intermittent faults.

DEFINITION 2. *A* **permanent mapping fault** *is a fault that continues to exist, unless some outside action takes place to remove its underlying cause.*

For example, any attempt to map between two unrelated concepts from two unrelated ontologies, i.e., two ontologies from different domains, will result in a permanent fault. This situation will continue indefinitely, unless, e.g., a change is made in the mapping semantics linking the ontologies.

DEFINITION 3. *A* **transient mapping fault** *is a type of fault that appears once, and remains in place for a short period of the time.*

A transient fault may corrupt the data of a system, but the system will remain operational. It is a statistical fault, and it is hard to predict when it will exactly happen. For example, the change of a company's stock symbol can result in a transient semantic mapping fault, if either the propagation of the change notification to related peers or applications is delayed, or the related peers or applications are unable to incorporate the change immediately.

DEFINITION 4. *An* **intermittent mapping fault** *is a fault that occurs periodically. It appears for a short period of time, disappears, and then reappears repeatedly.*

For example, in a situation where ontology modification is not a full substitution of one ontology by another, i.e. mapping is only partially corrupted, it is possible for the related peers to continue interact, but with possibility of repeated faults, *intermittent type of fault*

Fault type definitions can be formally represented as follow:

$$f(m) = \begin{cases} Tr, & \text{for } t = t_1 \\ Int, & \text{for } t = t_i\text{: } i = [1..n],\ \sum_{i=1}^{n} t_i < T \\ Pr, & \text{for } t = t_1, t_2, ..., T \end{cases}$$

where Pr , Tr and Int stand for permanent, transient and intermittent fault types respectively, t_i is the duration or the period in which mapping is corrupted , and T is an entire system operation duration.

3. MAPPING FAULTS DUE TO ONTOLOGY MODIFICATION

In this section we describe different scenarios in which different types of semantic mapping faults occur as a result of ontology modification, i.e. the process of replacing or updating an old ontology with a new one[2]. In the described scenarios, the emphasis is put on two aspects of modifications: i. the *ontology modification extent*, i.e. level of modification, and ii. *modification update message*. We first start by describing ontology modification forms.

3.1 Ontology Modification Forms

Ontology modification occurs in various forms including concept or datatype modification[22]. Some forms of ontology modification based on concept modification are listed below:

• *adding new concepts* to the existing ontology. For example, adding a newly discovered class or type of drugs, proteins or diseases to the existing relevant ontologies.

• *deleting concepts* from existing ontologies. For reasons such as outdated concepts, no longer used or useful concepts, concepts may be deleted from the ontology structure.

• change in *meaning/ conceptualization,* of the existing concept. The change could take the form of removing/adding concept relation or concept property. For example, attaching hydrogen fuel type to the car concept, the fuel type which does not exist previously, is an example of change in concept conceptualization by *adding* new property. Removing disc drive from personal computer (PC) concept because the company no longer builds PCs with disc drive build-in is another example of the change in the ontology concept by *removing* the concept property.

3.2 Mapping Fault Scenarios

A short description of faults that could be caused by ontology modifications are listed below.

• As noted above, in circumstances where ontology modification is not a complete substitution to the previous ontologies, there is a possibility for related peers or application to continue working. In the described scenario, there are possibilities for fault occurrence -*intermittent type of fault.* Faults will occur because there are situations where related peers

[2]We use ontology modification for both ontology versioning and evolution. This is because both introduce modification to the existing ontology, the subject of concern from fault study perspective.

are unable to interpret the meaning of concepts in modified ontologies.

• The *level* of ontology modification and whether or not the modified concepts will be *used* in the mapping process will determine the mapping result. The higher the level of modification and the repeated use of the modified concepts could give rise to inability of related applications or peers to work with the modified ontology -*permanent type of fault*.

• The process of ontology modification could result in one of the following two situations:

* *unavailability* for short periods of time, if the access to the ontology is blocked while the modification is performed or,

* a *race* situation between information source and information users, if the ontology user is informed about the change *before* or *after* the modification is made. Each of the two described situations will result in the *transient type of fault*.

For a detailed list and analysis of fault causes as well as the relation between fault causes and fault types we encourage the reader to see [18][19].

4. CASE STUDY: EMERGENT SEMANTICS

Emergent behavior is a well-known phenomenon in biology, physics and (distributed) computing. For example, several optimization and network routing techniques have been inspired by the way the behavior of an ant colony, as a whole, emerges from local interactions between individual ants.[3] Similarly, local cooperation between robots in multi-robot systems for search and rescue operations has been modeled after the formation of flocks of birds [5].

Inspired by emergent behavior, the approach of Emergent Semantics has been proposed as a solution to the semantic interoperability problem among autonomous, heterogeneous information sources with local ontologies, SP2P systems.

Emergent Semantics refers to the bottom-up construction of interoperable systems, in which semantically related peers are discovered and linked together during normal operation of the system, as part of regular search and query forwarding operations. In this process, individual information source providers supply semantic mappings between their own local and semantically-related foreign information sources, and adjust their local mappings based on query results [1][2][11][16][29]. This process can be conceived as finding minimum common knowledge problem among all peers' contextual ontologies in the network.

The straight forward step in the emergent shared semantics among heterogonous and distributed local ontologies is that, every-time a local peer P encounters another peer $\bar{P}$ that could provide correct answer to its query, i.e. a peer with comparable semantic representation, the existing semantic mapping between two peers P and $\bar{P}$ will be further reinforced. Conversely, if the returned answer is not satisfactory the mapping between the two peers is weakened.

The described steps are conceptualized as the directed graph creation procedure where semantically comparable peers are discovered and linked to each other through normal operation of the system. The end result graph will encompass a certain percentage of the total peers in the network.

The fundamental requisite for the creation of the described semantic graph is the existence of the local mappings between peers possessing different representations of the domain under discourse and the correctness of the local mappings. Fig.1 is a graphical representation of such procedure where the filled peer is a query initiator, the links represent a directed mapping from source to target and semantically related peers are connected by a directed link.

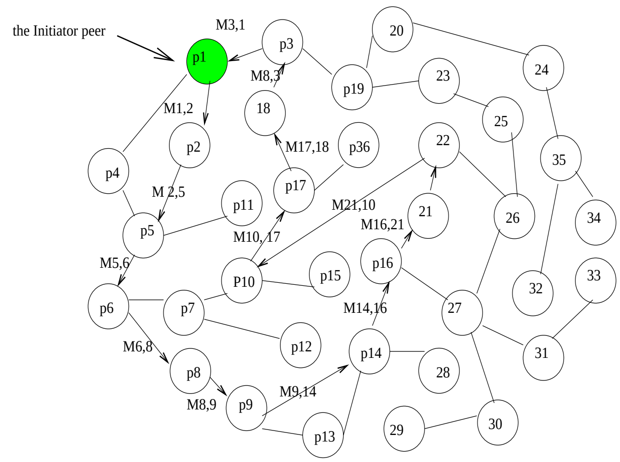

Figure 1: related peers without temporal fault handling

We believe that the problem somehow starts when peers are unable to answer queries or provide correct answers. In other words, the distinction between permanent and non-permanent mapping faults is a subtle issue when correct answers to the queries are not obtained. Not making distinction between permanent and non-permanent semantic mapping faults would result in the erroneous labeling of peers with the incompatible knowledge representation.

The consequence of erroneous labeling peers is determined by the number of outgoing mappings links each peer has in the network. We will consider two cases:

Case1: In this case, one of the peers on the mapping path has only *one outgoing link*. This case is represented by Fig. 2, where peer P_1 is the query initiator peer, peer P_k is the peer with one outgoing link M_k and all links from peer P_{k+1} are different paths returning results to the initiator peer P_1. Little circles on the paths represents different peers who participated in forming the results. It is clear from the Fig. 2 that unless the system has the ability to distinguish between transient and permanent mapping faults, if M_K, the mapping between Peer P_k and peer P_{k+1} gets wrong, even just for a short period of time, the peer P_1 will conclude that the out-going mapping link M_1 is not totally reliable. This is because even a temporary failure of one mapping link M_k, will result in all paths (passing through M_k) returning incorrect responses. And, given that P_1 has only local knowledge, it will conclude M_1 to be incorrect.

Based on: **i.** the state of the link M, i.e., its prior value, and **ii.** repeated rate of the transient fault, peer P_k and all other peers on the mapping paths go through peer P_k only, could be excluded from participation in emerging shared semantics.

Case2: In this case, we are considering a situation when peers have k outgoing mapping links and k > 1. Fig. 3 represents this case. It shows that P_k has three out-going mapping links { M_{k1}, M_{k2} ,M_{k3} }. Hence, the decision on the reliability or trustworthiness of the out-going link M_1

[3]http://iridia.ulb.ac.be/∼mdorigo/ACO/ACO.html

does not depend entirely on the out-going link M_k as it was the situation in the case1. Nevertheless, not distinguishing between transient and permanent mapping faults, i.e., treating the mapping link M_k as permanent faulty mapping, will have, depending on the result evaluation function, a negative impact on the perception about the correctness of the outgoing mapping link M_1.

The wrong perception about any out-going mapping link, when peers have k out-going links, could impact the way subsequent queries will be routed. Going back to the Fig. 3, if the origin trust in the outgoing link M_1 and M_2 were x and y value respectively, were $x - y = d$, a transient fault on the mapping link M_{k1} downgraded the trust value of M_1 by any value $\geq$ d then, Peer P_k will favor M_2 over the M_1 for next query forwarding. This action in turn could isolate certain number of peers which in turn jeopardizes the completeness of emerging semantic.

The above two cases convince us to believe that a complete semantic emergence between independent and heterogeneous ontologies is not possible without tolerating semantic mapping faults. In the next section will present a solution to tolerate a non-permanent mapping fault(s).

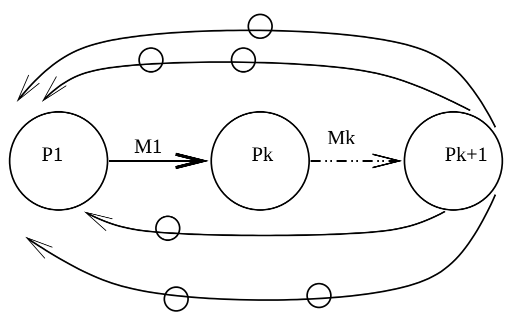

Figure 2: a peer on the mapping path has one out-going link

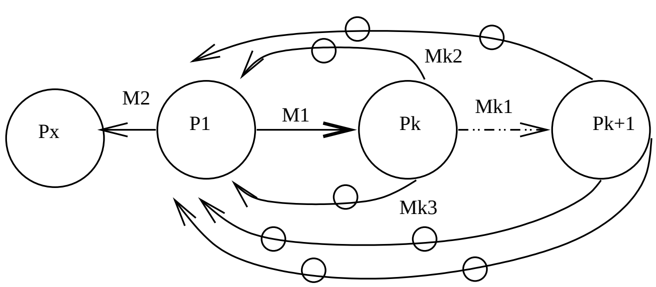

Figure 3: peers on the mapping path have multiple outgoing links

5. PROPOSED SOLUTION

There are various fault recovery solutions including checkpoints, rollback, error log analysis, etc. Some of the mentioned solutions are difficult to implement in P2P networks and others are not appropriate for use in context of the semantic mapping. We propose a solution to detect the semantic mapping faults which is simple in concept, easy to apply. It is based on the well known and proven correct technique in hardware (sofware) domain - the *time redundancy* technique [3][4][10]. The proposed solution is made of the following two steps:

1) To detect faults, peers will be tested with a repeated query as follows:

   a) Submit K sequential queries in place of one query every time query submission or query forwarding is performed. Queries are separated from each other by a time Δ. For instance, if K is set to 2 then the origin query and its clone will be separated by Δ time. That is, the second query will be posed at $t_0 + \Delta$, where t_0 is the time for initial query and Δ is the delay time between the two sequential queries. The system designer determines the maximum transient-pulse duration Δ that the system must tolerate.

   b) Query answers from replicated queries are compared for consistency. The inconsistency among answers for the same query is a deciding criterion for the transient fault occurrences. The consistency checking leads to the following two cases:

      (i) If query answers were consistent and incorrect then querying peer concludes that the queried peer is incapable of providing an answer to the query. Hence, it is *permanently faulty* relative to the posed query.

      (ii) If query answers were inconsistent, then a *transient fault* must have occurred, and an action should take place to eliminate its negative impact.

   The consistency relation is a system defined relation. Examples of relation consistency between two answers A_{s1} and A_{s2} is { $\sqsubseteq$, $\sqsupseteq$, $\equiv$ }, when answer $A_{s1} \sqsubseteq A_{s2}$ means that A_{s1} is less general than A_{s2}, $A_{s1} \sqsupseteq A_{s2}$ means that A_{s1} is more general than A_{s2} and $A_{S1} \equiv A_{s2}$, means that both A_{s1} and A_{s2} are exactly the same.

2) A transient fault recovery action comprises two steps:

   a) Query answer cancellation. If a transient fault is detected, the infected query impact on the semantic relation between peers should be ignored. This is achieved by sending a *cancel* signal to the peer that originally initiated the query. The cancel signal has one parameter, a *query-id.* The query-id identifies the query for the semantic mapping under investigation. As each peer returns the cancel signal to the peer it received the query from, the signal reaches the query initiator and the result of a query with the query-id in the cancel signal will be ignored. The result of the query will be ineffective on the grounds of the trust peers have in their outgoing links. Variations of this step is disused in [19])

   b) Query re-submission. In order for queries to recover from the impact of the transient faults, query resubmission needs to take place. This happens after waiting for Δ_2 length of the time from the last time a transient fault is detected and query re-submission could take place. The query re-submission can be repeated up to Ktimes. The Δ_2 value and the *number of query retry* are system parameters. These values

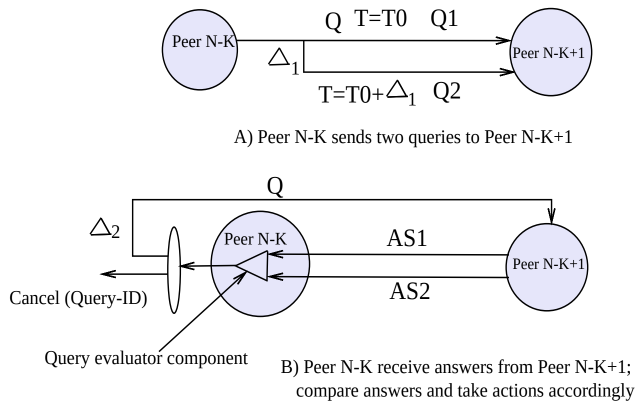

Figure 4: Query replication and answer consistency verification

will be set by system administration in such a way that a system will maximize the recall for the least additional queries. These values could be determined experimentally. Further, [17], suggests that ontology update notification in distributed systems should be enforced and performed within a time window. In the latter case the Δ_2 value will be set equal to time constraint.

Fig 4 is a graphical representation of our proposed solution when K = 2 and delay between queries is Δ. Having each peer checking and capturing transient mapping faults, we will build a robust system where chances for expelling peers for the non-permanent semantic mapping faults, hence bringing down chances for emerging complete ontologies, are minimized.

6. ILLUSTRATIVE EXAMPLE

In this section, we will demonstrate our solution through an example about emerging shared semantics for laptop ontology. Concepts used for modeling Laptops independently by Future shop, Sony, Bust Buy and Ebay have been used in the example. Two partial instances of store specific ontologies, [4] are provided below.

```
< owl : Ontology   rdf : about = ""/ >
  < owl : Class   rdf : ID = "laptop"/ >
  < laptop   rdf : ID =" 12356789"
    < name   rdf : datatype =
    "http : //www.w3.org/2001/XMLSchema#string"/ >
    SZseries < /name >
    < price   rdf : datatype =
    "http : //www.w3.org/2001/XMLSchema#currency" >
    $1999.99 < /price >
    < features   rdf : datatype =
    "http : //www.w3.org/2001/XMLSchema#string" >
    1.5   GHZ < /feature >
    < make   rdf : datatype =
    "http : //www.w3.org/2001/XMLSchema#string" >
    Sony < /make >
  < /laptop >
```

[4]Because the space limitation we provided only two instances of ontologies.

```
< owl : Ontology   rdf : about = ""/ >
  < owl : Class   rdf : ID = "notebook"/ >
  < notebook   rdf : ID =" 12334br456"
    < brand   rdf : datatype =
    "http : //www.w3.org/2001/XMLSchema#string"/ >
    TecraSeries < /brand >
    < price   rdf : datatype =
    "http : //www.w3.org/2001/XMLSchema#currency" >
    $999.99 < /price >
    < qualities   rdf : datatype =
    "http : //www.w3.org/2001/XMLSchema#string" >
    AMD Turion 64 X2 Duo Core TL-50< /feature >
    < tradeName   rdf : datatype =
    "http : //www.w3.org/2001/XMLSchema#string" >
    Toshiba < /tradeName >
  < /notebook >
```

Fig. 5 and 6 represents two different SP2P settings among four stores. Tables 1, 2 and 3 enable semantic mappings for concepts used in the sparql query provided below.

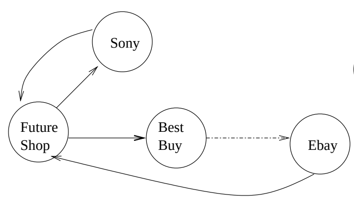

Figure 5: a peer on the mapping path has one outgoing link

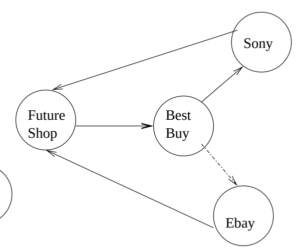

Figure 6: peers on the mapping path have multiple out-going links

For the purpose of the mappings, we assume the following logic relations are implemented in mapping procedure: $\{\equiv, \sqsubset, \sqsupset, *, \perp\}$ where, $c_1 \equiv c_2$ means that the two concepts are synonyms. We consider semantic affinity between synonyms concepts to be 1.0. The relation $c_1 \sqsupset c_2$ means c_1 is hypernym of c_2. The relation $c_1 \sqsubset c_2$, means that the c_1 have a hyponym relation to c_2. A hyponym is the opposite of a hypernym. The semantic affinity for hypernym and hyponym is set to 0.5. The relation $\perp$ means that two concepts have no semantic relation with each other. The semantic affinity between two un-related concepts is set to 0.0. Any other relations between concepts other than those described above will be captured by $*$ relation. The semantic affinity between concepts having $*$ relation is set to 0.25.

```
Q   =
PREFIX com < http : //www.../ ~#computer/ >
SELECT ?Operating System ?Display ?Weight
FROM < http : //www.../ ~/computer.owl/ >
WHERE
{
   ?laptop com:Operating System ?Operating System;
         com:Display ?Display;
         com:Weight ?Weight.
   FILTER (com : Price < 1000).
}
```

Future Shop	**affinity type**	**BestBuy Store**	**affinity ratio**
Laptop	$\equiv$	Notebook	1.0
Operating System	$\sqsubset$	Software	0.5
Display size	$\sqsubset$	Screen size	0.5
Weight	$\sqsubset$	Dimension	0.5

Table 1: FShope→BestBuy

Best Buy	**affinity type**	**Ebay**	**affinity ratio**
Notebook	$\equiv$	Notebook	1.0
Dimension	$\perp$	null	0.0
Screen size	$\sqsupset$	Screen size	0.5
Software	$\sqsupset$	OS	0.5

Table 2: BestBuy→Ebay

Future Shop	**affinity type**	**Sony**	**affinity ratio**
Software	$\sqsupset$	System Software	0.5
Display	$\sqsupset$	Display	0.5
Dimension	$\sqsupset$	Description	0.5

Table 3: FShope→Sony

Using above local ontologies, network settings, point to point mappings and query, we will now study the effect of our proposed solution on eliminating/reducing the destructive effect of the mapping faults on the emerging shared semantics. We will do that by walkthrough a query forwarding steps on two different system settings: **i.** a system without fault-tolerant capability and, **ii.** a system with built-in fault-tolerant capability. The outcome of the running example on the two systems will be compared to determine the effect of not handling the non-permanent mapping faults on the Emergent Semantics.

6.1 One Out-going Mapping Link and no Fault-tolerance

Having a query Q been posed on a Future shop store in a network of stores depicted by Fig. 2, concepts included in the query (Laptop, Operating system, Display and Weight) will be translated along a mapping path Fshop → BestBuy →Ebay and, results will return to Future shop store.

Lets assume that BestBuy does not have a Laptop which would satisfy the query constraint, i.e. a laptop with value less than a $1000, then it will provide no answer to the Future shop store; it will forward the query to its *only* semantically related store, the Ebay. Future shop store will receive Laptop information from the Ebay, this is because the semantic relation along the translation path for the Laptop concept (Laptop → Notebook → Notebook) has been preserved and its equivalent to 1.0. Other product attributes included in the end result product description are: *System Software* with semantic relation equal to 0.5 (Operating System → Software → System Software) and *Screen size* with semantic relation equal to 0.5 as well (Display→Display→Screen size).

It is clear from comparing the list of product attributes included in the query to the list of attributes in the result product description that one of the attributes, the *weight* attribute, has been dropped by the translation process. Once the Future shop store starts calculating semantic similarity for the query result, the value of the weight attributed will be set to 0.0. The semantic similarity vector for query result will contain the following values: $\boxed{1.0 \mid 0.5 \mid 0.5 \mid 0.0}$ for concepts: Laptop, Operating System, Display and Weight respectively.

For the purpose of query answer validation, we consider an answer to be correct if: **i.** it satisfies the query constraints and, **ii.** the SUM of semantic relation values for concepts returned in answer description over a number of concepts be $\geq k$, where k is a system defined value. For instance, If k value were set to ≥ 0.5, then query result returned by Ebay will be considered correct. The newly discovered concepts in the answer will be incorporated in the query initiators local mapping, i.e. local mapping is adjusted (new concepts integrated into local mapping), and believes in correctness of the mapping link Fshop → BestBuy will be increased.

The described situation assumes a perfect world. In other words, no *changes or modification neither to mappings nor to local ontologies* during query process, *perfect peers behavior*, *continues service availability* and probably *no use of temporal concepts or data* is assumed.

Unfortunately, the world is not perfect. As soon as one of the listed conditions does not hold, the above described scenario will not work properly. For example, if it happens that Ebay is blocking the incoming queries temporarily- while it updates its local ontology, then the answer to the query Q will return no result. Depending on the current value of the mapping link Fshop→BestBuy, this mapping link could be labeled as faulty link. Thus, resulting in *permanent disconnect between Fshop and sub-network starts from BestBuy.*

6.2 One Out-going Mapping Link with Fault-tolerance

Having our solution in place, lets consider the situation described in the subsection 6.1. BestBuy will submit query Q twice instead of one to Ebay. The outcome of two queries will be checked by BestBuy for detecting query result inconsistency. Since Ebay was blocking the incoming query only temporarily, two different query results will be returned to BestBuy and temporary faults detected. BestBuy will send *cancel* message with query id to the Future shop store. The trans message results in ignoring the effect of query Q answers on changing believes in correctness of out-going mapping link Fshop→BestBuy. After waiting for a Δ_2 time, the system will re-submit the query Q and start normal operation. In other words, having our solution in place, we will prevent the system from equally treating temporary and permanent mapping faults and, now we could have more confidence in achieving our goal -reaching shared understanding of semantics of concepts in ontologies.

6.3 Multiple Out-going Mapping Links without Fault-tolerance

In this case we want to study the effect of transient mapping faults in situation when peers have more than one mapping links to others. Building multiple mapping links is costly but, depending entirely on one mapping link is not trouble free as well. We have illustrated the later case in subsection 6.1.

Figure 3, represents the discussed example in subsection 6.1 with BestBuy having more than one mapping link, i.e., two mapping links. Following query forward steps described in subsection 6.1, once BestBuy receives query Q from Future shop, it will forwarded it to its semantically related peers, i.e., Ebay and Sony store. Assuming Ebay blocking incoming queries temporarily, Future shop will receive a query vector result from Sony store for concepts included in the query $\boxed{1.0 \mid 0.5 \mid 0.5 \mid 0.5}$ [5] and null value from Ebay. It is obvious that the null value returned from Ebay will have negative impact on evaluating the correctness of the out-going mapping link Fshop→BestBuy. The intensity of this negative impact depends on: **i.** the other result returned from the path (Fshop→BestBuy→Sony→Fshop) and, **ii.** the way returned results are used in evaluating the correctness of the out-going mapping link (evaluation function). In general we will have the following three cases; other sub-cases are also possible:

1. if the result returned by mapping path (e.g. Fshop → BestBuy → Sony → FShop) is not correct, then the decision about trustworthiness of the out-going mapping link Fshop→Ebay will depend entirely on the result returned from Ebay. Hence, the temporary failure will have drastic effect, i.e. the problem is transformed to the previous case, and there will be a possibility of total abandoning of the out-going mapping link. This is not the case in our example.

2. if the evaluation function is based on taking median or average result values -the average value is used in our case, then the transient fault will have severe impact and, it could result in abounding out-going mapping link. This case is applied to our example.

3. if out-going mapping link continues to be used as long as one correct answer is returned, i.e. the required correct value for using an out-going mapping link set to be > 0, then the temporary mapping fault will have limited effect on the validation of the out-going mapping links unless, the only link which returns correct result infected by temporary fault.

Having our proposed solution in place to detect and eliminate semantic mapping faults we are going to have a denser graph than the one presented in the Fig.1. This is because all the peers, which were excluded from participation in the first place because of the temporary semantic mapping faults, will be included this time. Fig. 7 represents a picture of our new graph where the new filled peers are the new peers which have been added to the first graph.

7. RELATED WORK

Recently some researchers have started looking into the issues that one could characterize as pertaining to transient mapping faults. For instance, in [31], peers that participate frequently in answering queries but their answers are not correct are double checked for schema changes.

Acknowledging the fact that P2P networks could change during query propagation, i.e., a peer may become temporarily unavailable or a new peer with relevant information source joins the network, Zaihrayeu [30] highlights three different scenarios which have the potential for generating transient faults.

In an effort to manage dynamic versioning and evolution

[5](Laptop → Notebook → Notebook), (Operating System → Software → System Software), (Display→Display→Screen size) and (Weight→Dimension→ Description)

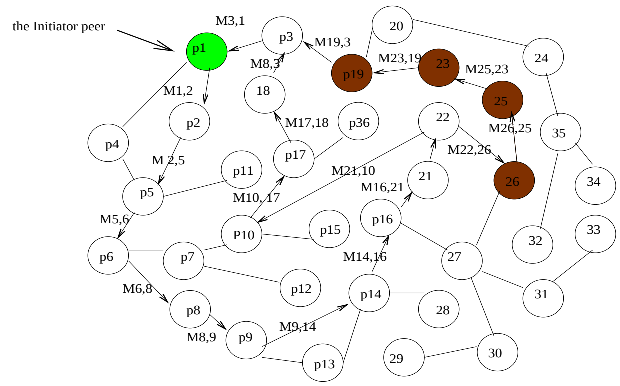

Figure 7: related peers with temporal fault handling

of distributed ontologies, Ma, et al. [17] argue for an urgent need for an ontology evolution representation approach with time constraints. While we concur with Ma, et al. that inconsistency among connected ontologies due to evolution and versioning has to be prevented, we do not believe that requiring sources which are undergoing changes to inform their connected ontologies (or risk isolation) will help to solve the problem of transient mapping faults. This is because the target ontology will take time to respond to the update notification and as such during the intervening period (i.e., between notification and update) ontologies will remain inconsistent and so transient mapping faults will arise.

McCann, et al. [20] build a MAVERIC (Mapping Verification) system which continuously monitors sources *"for detecting broken mappings"* automatically. In this system, information sources are probed periodically and query answers are compared to the prior known answers. Once newly retrieved query answers differ from the predicted/existing ones, an alert signal about a potential broken map is sent to the system administrator. This work is relevant to our study but differs from the approach we take. First, while McCann, et al. focus on faults due to information source changes, we consider a broader set of fault sources including not only changes in information sources but also on source unavailability, static mapping, etc. Second, instead of monitoring information sources continuously, we suggest detecting changes only when we query information sources. That is, we are interested only in detecting changes that are current and relevant to our queries. Further, they have not fully considered the implications of frequent and numerous changes of information sources.

We concur with Colazzo and Stariani [9], that corrupted mappings have drastic consequence on the query results and optimization techniques depend on them. However, our solution to the problem is different. Our algorithm differentiates between permanent and transient mapping faults. We tolerate transient mapping faults and detected the permanent ones. Our solution, leaves to system administrators to deal with permanent faults. This is in order to not lose algorithm's general applicability. In contrast to Colazzo and Stariani approach, our solution is data representation and query language independent. That is, it could be used as plug-in with any SP2P systems. This is an important future, since data representational and query language dependent solution has limited expediency.

8. CONCLUSION AND FUTURE WORK

In this work, we have identified that one of the shortcomings of the existing works related to semantic mapping faults is the deficiency to distinguish between permanent and non-permanent semantic mapping faults. A formal definition of the semantic mapping fault along the temporal dimension, the destructive effect of the temporal semantic mapping faults on the Emergent Semantic and, a solution to minimize the effect of the non-permanent semantic mapping fault also have been provided. Through an example we demonstrated that *it is not possible to reach complete state of Emergent Semantics among independent, heterogeneous and distributed local ontologies unless there is a built-in capability in the system to eliminate/reduce temporary semantic mapping faults*. Our proposed solution enables minimizing the effect of transient and intermittent faults. Hence, it contributes in emerging condition for a more agreeable semantics. This is because, a higher number of peers in the network will participate in emerging semantics. Our immediate future work is to complete the prototype implementation.

9. REFERENCES

[1] K. Aberer, P. Cudre-Mauroux, M. Hauswirth. Start making sense: The Chatty Web approach for global semantic agreements. In *Journal of Web Semantics*, 1(1): 89-114, 2003.

[2] K. Aberer, T. Catarci, et.al. Emergent semantics systems. In *Semantics of a Networked World. Semantics for Grid Databases. First International IFIP Conference, ICSNW*, P: 14-43, 2004.

[3] L. Anghel, D. Alexandrescu, M. Nicolaidis . Evaluation of a soft error tolerance technique based on time and/or space redundancy. In *Proc. of 13th Symposium on Integrated Circuits and Systems Design*, P: 237 - 242, 2000.

[4] A. Avizienis. The Methodology of N-Version Programming. *Chapter 2 of Software Fault Tolerance, M. R. Lyu (ed.)*, Wiley publishing, 23-46, 1995.

[5] E. Bahceci, O. Soysal, and E. Sahin A Review: Pattern Formation and Adaptation in Multi-Robot Systems. Report CMU-RI-TR-03-43, Robotics Institute, Carnegie Mellon University, 2003.

[6] M. Bonifacio, P. Bouquet, G. Mameli and M. Nori. Peer-Mediated Distributed Knowledge Management. In *proc. of International Symposium AMKM*, p: 31-47, 2003.

[7] P. Bouquet, A. Donk, et al. ConTcXtualizcdlocal ontology specification via ctxml. In *AAAI workshop on Meaning Negotiation*, 2002.

[8] S. Castano, A. Ferrara, S. Montanelli. H-Match: an Algorithm for Dynamically Matching Ontologies in Peer-based Systems. In *Proc. of the 1st VLDB Int. Workshop on Semantic Web and Databases (SWDB)*, P: 231-250, 2003.

[9] D. Colazzo and C. Sartiani. Mapping Maintenance in XML P2P Databases. In *Proc. Tenth International Symposium on Database Programming Languages (DBPL)*, Pages 74-89, 2005.

[10] E. Dupont, M. Nicolaidis and P. Rohr. Embedded Robustness IPs for Transient-Error-Free ICs. In *IEEE Design & Test* 19(3): 56 - 70, 2002.

[11] P. Fergus, A. Mingkhwan, et. al. Distributed emergent semantics in P2P networks. In *Proc. of the Second IASTED International Conference on Information and Knowledge Sharing*, P: 75-82, 2003.

[12] R. J. Glushko and T. McGrath. Document Engineering. The MIT Press,2005.

[13] T. R. Gruber. The Role of Common Ontology in Achieving Sharable, Reusable Knowledge Bases. In *Proceedings of the 2nd International Conference on Prin- ciples of Knowledge Representation and Reasoning*, 601-602, 1991.

[14] N. Guarino. Formal ontology and information systems. In *Proceedings of Formal Ontology in Information Systems*, P. 3-15, 1998.

[15] A. Halevy, Z. Ives, P. Mork, and I. Tatarinov. Piazza: Mediation and integration infrastructure for semantic web data. In *proceedings of the International World-Wide Web Conference WWW-03*, 2003.

[16] Larry K. Hanjo J. & Wooju K . Emergent Semantics in Knowledge Sifter: An Evolutionary Search Agent Based on Semantic Web Services. In *Journal on Data Semantics VI*, 187-209, 2006.

[17] Y. Ma, B. Jin, Y. Li et al. A Timing Analysis Model for Ontology Evolutions Based on Distributed Environments. In *Zhi-Hua Zhou, Hang Li, Qiang Yang (Eds.): Advances in Knowledge Discovery and Data Mining, 11th Pacific-Asia Conference, PAKDD, LNCS 4426*, pages 183-192, 2007.

[18] A-R. Mawlood-Yunis, M. Weiss, N. Santoro. Fault Classification in P2P Semantic Mapping. Workshop on Semantic Web for Collaborative Knowledge Acquisition (SWeCKa) at International Conference on Artificial Intelligence (IJCAI), 2007.

[19] A-R. Mawlood-Yunis, M. Weiss and N. Santoro. Fault-tolerant Emergent Semantics in P2P Networks. Book Chapter in *Semantic Web Engineering in the Knowledge Society*, Idea Group, 2008 (to appear).

[20] R. McCann, B. AlShebli et al. Mapping maintenance for data integration systems In *Proceedings of the 31st international conference on VLDB*, Pages: 1018-1029, 2005

[21] E. Mena, V. Kashyap, et. al. Imprecise answers in distributed environments: estimation of information loss for multi-ontology based query processing. In *International Journal of Cooperative Information Systems*, 9(4): 403-25, 2000.

[22] M. Klein PhD Disseration, Vrije Universiteit, Amsterdam, 2004. *http://www.cs.vu.nl/∼mcaklein/thesis/*

[23] C.F. Naiman and A.M. Ouskel. A classification of semantic conflicts in heterogeneous database systems. In *Journal of Organizational Computing, 5(2):167-193*, 1995.

[24] N. F. Noy and M. Klein. Ontology Evolution: Not the Same as Schema Evolution. In *knowledge and Information Systems*, 6(4):428-440, 2004.

[25] D. K. Paradhan. Fault-Tolerant Computing System Design. *Prentice-Hall PTR* publication, 1996.

[26] J. F. Roddick. A survey of schema versioning issues for database systems. In *Information and Software Technology*,37(7):383-393,1995.

[27] L.M. Stephens, M.N. Huhns. Consensus ontologies. Reconciling the semantics of Web pages and agents. In *IEEE Internet Computing*, 5(5): 92-95, 2001.

[28] C. Tempich, S. Staab, A. Wranik. Distributed semantic query: Remindin': semantic query routing in peer-to-peer networks based on social metaphors In *Proc. of the 13th international conference on World Wide Web*, P: 640-649,2004.

[29] S. Staab. Emergent Semantics. In *IEEE Intelligent Systems*, 17(1):78-86, 2002.

[30] I. Zaihrayeu. Towards Peer-to-Peer Information Management Systems. PhD Dissertation, International Doctorate School in Information and Communication Technologies, DIT - University of Trento, 2006.

[31] H. Zhuge, J. Liu et.al. Semantic-based query routing and heterogeneous data integration in peer-to-peer semantic link networks. In *Semantics of a Networked World. Semantics for Grid Databases. First International IFIP Conference, LNCS 3226)*, 2004, p 91-107

[32] H. Zhu, S.E Madnick and M.D. Siegel. Effective data integration in the presence of temporal semantic conflicts. In *Proc. 11th International Symposium on Temporal Representation and Reasoning*, pages 109-114, 2004.