Coping with Diverse Semantic Models when Routing Ubiquitous Computing Information

Song Guo, John Keeney, Declan O'Sullivan, David Lewis

Knowledge & Data Engineering Group (KDEG)
Centre for Telecommunications Value Chain Research (CTVR)
School of Computer Science & Statistics, Trinity College, Dublin, Dublin, Ireland.
{gsong | John.Keeney | Declan.OSullivan | Dave.Lewis}@cs.tcd.ie

Abstract - **Routing of contextual information within ubiquitous computing environments is a key challenge that must be tackled for such environments to be successful. It is accepted that such routing systems need to cope with mobile and volatile sources and destinations of contextual information, and semantic-based publish subscribe systems have been proposed as a means to support this. However such systems typically assume a common semantic model underpinning the routing which limits their ability to cope with heterogeneity. In contrast, the authors have developed a semantic-based publish subscribe system that is unique in allowing several semantic models to support routing. It is known from previous work that the number of semantic models in memory directly impacts on performance of routing, and that different loading strategies are needed, the selection of which is influenced by: the characteristics of the semantic model itself, the applications using the system and the network environment. This paper however just focuses on the experiments that have been undertaken to determine the key semantic model characteristics that may influence the selection of which loading strategy to use.**

I. Introduction

This paper is motivated by the challenge of establishing a common service for delivering context information to context-aware applications central to pervasive computing. Pervasive computing aims to support vast volumes of context messages from a multitude of sensors embedded in the fabric of everyday life reporting upon location, temperature, sound levels, RFID sensing to name but a few. Any scalable context delivery system must ensure the accurate delivery of context events to the consumers that require them.

There are a number of major challenges in distributing such contextual knowledge. The heterogeneity of contextual information means elements increasingly need to gather and pass contextual information to and from elements of different and possibly unknown types. The rapid evolution of sensors, actuators and applications leads to uncertainty about the type of contextual information that an element will gather, provide, route or use in the future. A volatile or mobile peer set means that a given element will need to gather or route contextual information to or from a frequently changing set of elements. These issues will lead to a level of heterogeneity that prevents context consumers accurately forming queries to match possibly unknown forms of relevant context events.

To address these challenges we adopt the Publish-Subscribe (Pub-Sub) paradigm [1] for the distribution of context information. The elements requiring contextual information express an interest through a subscription which is matched to messages published by other elements holding that type of information as it changes. Pub-Sub systems are already used for loosely coupled communication in a variety of applications. However, existing Pub-Sub systems require agreements on message types between the developers of publishing and subscribing applications. This places severe restrictions on the heterogeneity and dynamism of the information elements that can be exchanged. One solution to this is a Pub-Sub system that filters events based on matching client subscriptions to message attributes rather than the full message type, a technique known as content based networking. Content-Based Networks (CBN) thus facilitate still looser coupling between producer and consumer applications than Pub-Sub. Several CBN solutions and prototypes exist, e.g. [2] [3] [4] [5] [6]. However, widespread CBN deployments have been slow to emerge. This is partly due to the difficulty in reaching a general compromise between the expressiveness of event types and subscription filters and the need both to match these efficiently at CBN nodes. This falls well short of supporting the heterogeneity and flexibility that ubiquitous computing elements and applications require. Selecting a more expressive language involves a difficult trade-off, since higher level features, e.g. set functions, introduce more complexity into a CBN node, and may only be of use to a subset of applications.

Increasingly, researchers are turning to the use of ontology-based semantics to address this issue. The standardisation of ontology languages by the Semantic Web initiative at the World Wide Web Consortium (W3C) [7] has spurred an increasing number of researchers to use ontology-based semantics to support interoperability in heterogeneous and evolving systems [8] [9] [10]. A CBN based on messages containing semantic mark-up and queries is potentially far more flexible, open and reusable to new applications. We call such a semantic-based CBN a Knowledge-Based Network (KBN).

In this paper we focus on the problem where semantically enhanced messages may have been created using different knowledge bases (ontologies) to describe those semantics. We first provide more background on knowledge based networking and outline a number of strategies to deal with this issue. The remaining sections identify and evaluate the semantic characteristics and semantic reasoning requirements that influence the selection of one of these strategies.

II. Background

A. Knowledge Based Network

Knowledge-based networking involves the forwarding of messages across a network based on some semantics of the data and associated meta data of the message content. In previous papers [11] [12] [13] we have described a semantic-based CBN called the Knowledge Based Network (KBN). Producers of knowledge express the semantics of their available information based on an ontological representation of that information. Consumers express subscriptions upon that information as simple semantic queries. This approach provides loose semantic coupling between applications, which is vital as new waves of applications increasingly rely on using the application information, context and services offered by existing heterogeneous distributed applications. The particular flavour [14] of KBN which is investigated in this paper is an extension of Siena [2], which is an implementation of CBN middleware.

The use of an ontology is the key factor for enabling the semantic description of knowledge provided, queried and being routed around the network in KBN. It allows communication and knowledge sharing among distributed applications, by providing a semantically rich description and a common understanding of a domain of interest.

Producers and consumers express the semantics of their publications and subscriptions according to some shared ontology. This same ontology is then used by the KBN routers to efficiently route publications towards subscribers that have lodged subscriptions that match those publications. However, given the rapid evolution and dynamism of many distributed applications, there is increasingly a desire to allow applications which were designed independently and using different information structures to communicate that information without the necessity of custom building gateways. This is especially true in emerging ad-hoc pervasive computing and autonomic environments. Therefore, in some cases it is unreasonable to expect that all of the knowledge producers, knowledge consumers and knowledge routers have previously agreed on a single semantic model.

B. Related Work

Currently, a number of solutions utilize ontology technology in Pub-Sub systems. S-ToPSS [17] is a semantic-aware content based network, it proposed three approaches to enhance subscriptions and events semantically, in order to make the existing centralized syntactic matching algorithm semantic-aware meanwhile keep efficiency of current event matching technique. Another ontological pub/sub system called Ontology-based Pub/Sub system is developed by [8]. Aiming to improve expressiveness of events and subscriptions, it uses RDF [30] and DAML+OIL [31] techniques to describe events and subscriptions, where events and subscriptions are represented as RDF graphs and graph patterns respectively. [18] developed an independent concept-based layer which is built between the notification service and the pub/sub applications to provide a high level interaction among applications, in order to tackle the problem of event interaction among heterogeneous applications. Furthermore, our previous work [19] demonstrated how through the use of ontology and ontology mapping techniques applications built according to different standards (CIM [32] and SMI [33] were used) could interchange fault alarms over a Content Based Network (Elvin) [3] using an ontology based approach. However all ontological Pub/Sub Systems introduced above use a single common ontology to provide a semantically rich description and a common understanding of a domain among their applications in comparison to the extended KBN which supports multiple diverse ontologies.

C. Semantic Mapping in KBN router

In a previous paper [15] we described how we have extended the KBN so that it can cope with the situation where applications may be using multiple diverse ontologies. The incorporation of semantic interoperability within the KBN routers means that applications that subscribe to information according to one ontology can expect to receive information published according to a different ontology, if there exists a mapping between the ontologies. This feature then lowers the barrier for participation by applications in any particular KBN. Although it will potentially increase the workload of an individual KBN router the impact of the extra processing is far outweighed by the benefits from enabling semantic interoperability between applications. However, where possible the dynamic loading, parsing and reasoning of new ontologies into the KBN router's knowledge base should be minimised since this merging of ontologies can be a particularly expensive operation [16], particularly where this operation may need to be performed in a number of routers in the network of KBN routers.

As discussed in the next section we have identified a number of different strategies to support semantic mappings between ontologies, while this paper focuses on the semantic criteria that influence the selection of a strategy to merge these mappings.

D. Mapping Strategies for KBN Router

If subscriptions or publications contains heterogeneous semantic content then an individual KBN router will occasionally encounter an unknown concept (or property) that is not described in its own knowledge base (routing ontology). When a KBN router encounters an unknown ontological concept it should browse its set of semantic mappings to determine if it is able to handle that unknown concept. Since this operation may need to be performed on-the-fly, and may be a potentially expensive operation, there exists a number of different strategies to do this searching and merging of mappings in an efficient manner. Currently there four strategies available to incorporate semantic mapping information into the KBN router's routing ontology as follows:

- **The "Every mapping file" Strategy:** allows the router to load all available mappings and imported ontologies into its routing ontology at once. This strategy maximises the exploration of mappings to tackle the unknown data problem.
- **The "Appropriate mapping file" Strategy:** the KBN router checks mapping files available and merges

appropriate mapping ontologies, which contain at least one concept used by the conflicting subscription or notification

- **The "Appropriate individual mapping" Strategy:** checks the mappings and merges only the appropriate individual mappings into the router's routing ontology rather than the whole mapping file as in the second strategy
- **The "Appropriate & reference" strategy:** this strategy is similar to the third strategy above, however, unloaded ontologies referred to in the mapping may also be loaded, depending on the combination of mapping relations found, and the operator to be applied to the unknown concept.

The following section focuses on the conditions that influence the selection of one of these strategies to incorporate semantic mappings.

III. Influences on Strategy Selection

Different KBN routers could store different routing ontologies along with different numbers of mapping ontologies. This can cause significantly different repercussions on the reasoning performance of a KBN router executing a specific strategy to deal with unknown data. For instance, the "**every mapping file**" strategy is well-suited for the routers which store a small number of mapping ontologies, whereas strategies that do not import some of the ontologies referenced by mappings are well suited for the routers with large number of ontologies. Furthermore, the strategies that import referenced ontologies are preferable to the large-scale environment where the occurrence of unknown data is high. It is noticed that in a small scale scenario, it may be possible to examine the application running over the KBN to statically determine which strategy is most appropriate. However, in a large scale deployment, or where the ontologies stored in KBN and applications using the KBN may change, then it is necessary to dynamically manage and adapt which strategy is most appropriate. Hence, different mapping strategies can be configured in different KBN routers depending on the characteristics of ontologies stored in each KBN router, the type of application operating over the KBN, and the network environmental state.

Firstly, ontology characteristics that may impact strategy selection at a router are: the size and complexity of the applications' ontologies, routers' ontologies, mappings and referenced ontologies; the number of imported ontologies in a mapping file; the ontological mapping operators used in the mapping files; and the concept's position in the ontology's class hierarchy tree.

Secondly, the application characteristics that may impact strategy selection at an individual router are: the rate of publications and subscriptions and their active/inactive duration; the fault tolerance capability of KBN to respond gracefully to an mapping failure (e.g. a mapping is missed so an unknown concept or property remains unknown so cannot be routed correctly); the tolerance of application to handle false positive or false negative subscription matches; etc.

Finally, the environmental states that may impact strategy selection at an individual router are: the network scale, where a KBN deployment can range from enterprise scale to internet-scale; memory resources of an individual router; and the number of mapping ontologies stored in each KBN router.

Given different possible mapping strategies, our recent research has focussed on identifying which of the ontology, application and environmental characteristics mentioned above will be important in influencing strategy selection and what that influence might be with a view to building a decision making component to support strategy selection. However, due to space constraints this paper will focus purely on identifying the **ontology characteristics** that will be important for strategy selection.

IV. Ontology Characteristics

An ontology consists of classes and properties. Classes describe the characteristics or concepts of individual things within the ontology, while properties describe relationships between or about things. The class hierarchy tree in ontology is a set of concepts with equivalence or sub-/super-class semantic relationships between them, thus it is organised as a class taxonomy. Due to the formal nature of how many ontologies are specified it is possible to perform some reasoning over the classes and their properties to correctly derive this class hierarchy (classification or TBox reasoning). The root node is semantically the most generic class; whereas the leaves are the most specific classes. A sub-class is said to be *subsumed by* its super-classes, while a class *subsumes* its subclasses.

From a state of the art survey [20] [21] [22] [23] most researchers have taken the number of classes, properties and individuals along with the languages that are used to describe an ontology, as ontology characteristics to evaluate ontology reasoning. The results of such research have shown that these simple ontology characteristics are reasonable indicators with respect to reasoning performance. For instance, the number of classes and properties of an ontology influences the time for a reasoner to compute TBox classification (arranging the classes and properties into their reasoned hierarchy), while the number of individuals determines the amount of ABox realisation (finding the types of an individual, in particular its most specific type). In our work however, we were interested in exploring a wider set of ontology characteristics given that the KBN router's reasoning performance will be influenced by having to cope with several ontologies as opposed to just one ontology at a time. In order to do this, we designed an experiment to explore what these ontology characteristics might be.

The first hypothesis for the experiment was that size and expressivity of an individual ontology is a good indicator of the reasoning overhead required for that ontology.

The second hypothesis was that we could predict or bound the reasoning overhead for a merged ontology from the reasoning overheads of its constituent individual ontologies. This would help us select appropriate strategy to efficiently tackle heterogeneous data.

The third hypothesis was that different ontological operators used in the mappings would not lead to significantly different reasoning overheads in a merged ontology. The mapping operators that we consider in this work are rather restrictive, i.e., that a class (or property) in one ontology be *equivalent to* or be a *sub-/super-class* (or property) in another ontology. For instance, given a mapping where the *equivalence* operator is applied to link two different classes in different ontologies the reasoning requirement for that merged ontology is similar to the situation where a *sub-class* mapping operator is used. If the mapping operator does impact performance then we may need to include it as a factor in our strategy selection.

The fourth hypothesis was that the positions of the classes within the class hierarchy trees of the constituent ontologies will have little effect on the reasoning overhead of the merged ontology. Figure 1 shows a concrete example: concept **a** (root node in ont1) and **y** (middle node in ont2) are the mapped classes in a mapping ontology **m1** that only contains this one mapping. Assuming there is another mapping ontology **m2** that only specifies that classes **a** (root node in ont1) and **x** (root node in ont2) are mapped, would **m2** have more or less reasoning overhead than **m1,** if the same ontological mapping operator being used?

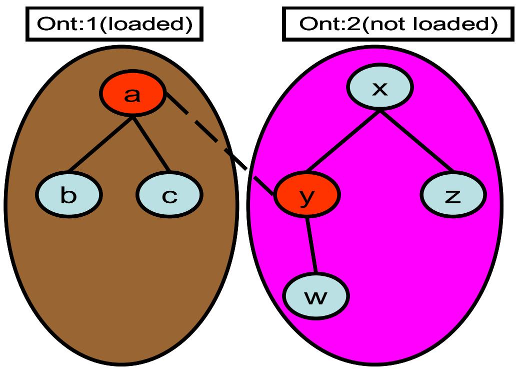

Figure 1. Illustration of classes positions in class hierarchical tree

V. Experimental Method

Three types of ontologies were used in the experiment: a **routing ontology** that is already loaded by a router and used to match and route publications and subscriptions; a number of **referenced ontologies** that contain definitions for classes or properties which are not described in the routing ontology; and a number of **mapping ontologies** that each contain semantic mappings between classes or properties in the routing ontology and referenced ontologies. When a mapping and a referenced ontology is incorporated into the routing ontology we use the term **merged ontology** to refer to the resulting ontology. We used ontologies from the semantic web research community as the routing and referenced ontologies in our experiment and manually created the mapping ontologies. These ontologies are discussed in section A below and the URLs for these ontologies can be found in Appendix 1. In the experiments, we measured the reasoning overhead of the ontologies individually and afterwards the reasoning overhead of the merged ontology. The reasoning overhead metrics used are described in section B. The experimental setup is discussed in section C.

A. *Test Ontologies*

1) *Routing and Referenced Ontologies used*

14 commonly available semantic web ontologies were used for our experiments. The selection of these ontologies was motivated by the fact that the ontologies are created by different people with diverse technical backgrounds. In this sense, the ontologies can be considered as representative of the natural range and diversity of ontologies that will be expected in a ubiquitous computing environment. Secondly, the ontologies were chosen in order to reflect a range of expressiveness[1] and ontology sizes, from ontologies with small number of statements to ontologies with large number of statements. [24] indicates that normally the smallest ontologies has less than 100 concepts, medium ontologies has between 100 and 1000 concepts, and large ontologies has more than 1000 concepts. However, in our experiment, these ontologies are categorised into four sets according to the number of statements that are the basic elements in ontology terminology, as the number of statements can more precisely reflect the size of ontology than the number of concepts. Table 1 summarises the ontologies according to the ontology characteristics of interest:

TABLE 1. INDIVIDUAL ONTOLOGY CHARACTERISTICS

Name	Number of Statements	DL Expressivity	Imports	Annotation
Teams	262	ALCIF	0	Small ontologies where number of statements is less than 1,000
Foodswapper	350	ALC(D)	0	
University	453	ALCR+OIF(D)	0	
Beer	576	ALHIF(D)	0	
Foaf	808	ALCHIF(D)	0	
Mad_cow	1,012	ALCHOIN(D)	0	Medium ontologies where number of stmts is between 1,000 and 5,000
Mindswappers	2,303	ALCHIF(D)	3	
Pizza	3,201	ALCF(D)	1	
Transportation	4,847	ALH(D)	0	
CongoService	5,199	ALCR+HOIF(D)	12	Large ontologies where num of stmts is between 5,000 and 10,000
Economy	5,489	ALH(D)	0	
Wine	5,710	ALCR+HOIF(D)	1	
MGED	14,501	AL(D)	0	Very large ontologies where num of stmts is greater than 10,000
Galen	64,673	ALCR+HF	0	

B. *Created Merged Ontologies*

15 example merged ontologies were manually created by the authors. For the purpose of comparing the reasoning overhead of each merged ontology with the reasoning

[1] Depending on the expressiveness of an ontology some of the following letters can be used to denote the presence of description logic features in the ontology, thereby capturing its reasoning complexity: AL - Attribute Logic: Conjunction, Universal Value Restriction, Limited Existential Quantification; C - Complement (together with AL allows Disjunction, Full Existential Quantification); R - Role Transitivity; H - Role Hierarchy; I - Role Inverse; O - Nominal; N - unqualified number restrictions; Q – qualified number restrictions; F - only functional number restrictions; (D) – Datatypes.

overhead of its constituent individual ontologies. An individual merged ontology has only two mapped classes and two constituent individual ontologies that are randomly chosen from the ontologies described above. The characteristics of the merged ontologies that were created are shown in Table 2:

TABLE 2. CHARACTERISTICS OF ONTOLOGIES MERGED USING MANUALLY CREATED MAPPINGS

Name	Number of Statements	DL Expressivity	Constituent Ontologies
FoodUniversity	695	ALCHOIN(D)	Foodswap & University
FoafFood	1,077	ALCHIF(D)	Foaf & Foodswap
FoafUniversity	1,157	ALCR+HOIF(D)	Foaf & University
BeerMadcow	1,485	ALCHOIN(D)	Beer & Mad_cow
FoafMadcow	1,716	ALCHOIN(D)	Foaf & Mad_cow
BeerMindswaper	2,774	ALCHIF(D)	Beer & Mindswappers
FoodPizza	3,460	ALCF(D)	Foodswap & Pizza
FoafPizza	3,901	ALCHIF(D)	Foaf & Pizza
BeerTransport	5,320	ALHIF(D)	Beer & Transportation
EconomyFood	5,374	ALCH(D)	Economy & Foodswapper
EconomyBeer	5,964	ALHIF(D)	Economy & Beer
EconomyTransport	10,239	ALH(D)	Economy & Transportation
EconomyCongo	10,599	ALCR+ HOIF(D)	Economy & CongoService
EconomyWine	12,336	ALCR+ HOIF(D)	Economy & Wine
EconomyMGED	19,882	ALH(D)	Economy & MGED

In order to investigate the effect of mapping operators and mapping positions upon performance, 10 mapping ontologies were created by altering some of the merged ontologies above to include different mapping operators and applying different mapping positions. The equivalence, sub-class, super-class mapping operators were all applied to map between classes in the two source ontologies.

In order to evaluate the impact of class position on reasoning performance, different classes from the top (T), middle (M) and bottom (B) of the class hierarchy trees were mapped using the *sub-class* ontological mapping operator. The class position we measured are shown as following:

- $CA_T \xleftrightarrow{Sub} CB_T$: *CA* and *CB* refer to classes in two different ontologies while the sub-script $_T$ means that the class is at the top (root) of the ontology's class hierarchy. $\xleftrightarrow{Sub}$ is the *sub-class* mapping operator used to link mapped classes
- $CA_T \xleftrightarrow{Sub} CB_B$: Here the sub-script $_B$ means that the class is at the bottom (leaf) of the ontology's class hierarchy.
- $CA_B \xleftrightarrow{Sub} CB_B$: Both mapped classes are leaf nodes
- $CA_M \xleftrightarrow{Sub} CB_M$: Here the sub-script $_M$ means that the class is at the middle of the ontology's class hierarchy. Both mapped classes are the intermediate nodes

C. *Metrics used for measuring Reasoning Overhead*

Based on previous work [13] [16] the following observations are of particular importance: loadtime reasoning in comparison to runtime querying is relatively expensive; the performance of different reasoners, and the reasoning load, will also change in a non-linear fashion depending on the size and expressiveness of the ontologies used and the level of ontology language used (e.g. OWL-Lite vs. OWL-DL) [21] [22] [25] [26] [27]. These observations are particularly important if ontologies are added or removed dynamically, as would be typical in a ubiquitous computing environment where applications will join and depart from the network. It was also observed that XML parsing time of the RDF was inconsistent and unpredictable and so was omitted from our metrics. Table 3 summarises the reasoning metrics used.

TABLE 3. PROVIDES A SUMMARY OF THE REASONING METRICS

Measure Metrics	Description
Loadtime without parsing time	Is given as the time taken for different reasoners to load, and check the ontology, combined with the time taken to perform TBox classification, perform ABox realisation and an initial query of all concepts
Loading time	The time takes to load ontologies into reasoner
Consistency checking time	Consistency checking ensures that an ontology does not contain any contradictory facts. In DL terminology, this is the operation to check the consistency of an ABox with respect to a TBox
Classification time	Classification can be defined as the computation of the subsumption hierarchy for classes and properties
Realisation time	Realisation finds the most specific classes that an individual belongs to, in other words computes the direct types for each of the individuals. It should be done after classification since direct types are defined with respect to a class hierarchy.
Concept querying time	First time to list all classes of an ontology
Runtime	The time taken to perform subsequent queries for the set of concepts in ontologies

D. *Experimental setup*

A previously implemented KBN router [14] was extended to implement all four of the mapping strategies discussed earlier in section section II.D. The Pellet reasoner [28] version 1.3. beta was embedded into the KBN router. Jena [29] was used throughout to access the ontologies and to measure the reasoning performance of Pellet. In order to minimise the adverse effect of inconsistent network connection speeds on reasoning performance all tested ontologies and their imported ontologies were cached locally on the machine running the tests. All tests were untaken on a Dell Inspiron 9300 laptop with 1.73 GHz Intel processor, 2GB of RAM, running Windows XP Service Pack 2. For Java-based tools, Sun's JDK 1.6.0 was used. All tests were run at least 20 times to provide statistically appropriate averages.

VI. EVALUATION

To address our first hypothesis we reasoned the selected source ontologies to see how reasoning performance was dependent on both the number of statements and the expressivity of the different ontologies, as shown in section A below.

To test our second hypothesis, we compared the reasoning overhead of a merged ontology with the combined reasoning overhead of its constituent individual ontologies, and this evaluation is discussed in section B. Here we examined if we could predict the reasoning overhead of a merged ontology given the reasoning overhead of the ontologies that made up that merged ontology.

Our third hypothesis was that the mapping operator used had little effect on the reasoning overhead of a merged ontology. Here we altered operators within mappings, and the results are presented in section C.

The findings from examining our fourth hypothesis, which stated that the hierarchical position of the classes used in mappings had little effect on the reasoning overhead of the merged ontology, are discussed in section D. Here we mapped classes from different positions within the class hierarchy of their individual ontologies and compared the reasoning overheads of the resulting merged ontologies.

A. Individual Ontology Reasoning

Figure 2 presents the reasoning overhead calculated on the individual ontologies. In Figure 2 the ontologies are arranged from smallest to largest (left to right) with the number of statements in each ontology given in parenthesis after its name. As can be seen from the times to load and reason and runtime performances of the different ontologies in Figure 2 (and with reference to Table 1), the reasoning overhead was greater for the larger ontologies. This confirms that reasoning performance is tied to the number of statements. However it was also observed that, although wine ontology is not the biggest one in size, it has the largest reasoning overhead. This is because it has the most complex structure and DL expressivity (see its column in Table 1) among tested ontologies. Given these reasoning times and the ontology DL expressivity information, an empirical finding is that both DL expressivity and ontology size impacts on reasoning performance, not just ontology size.

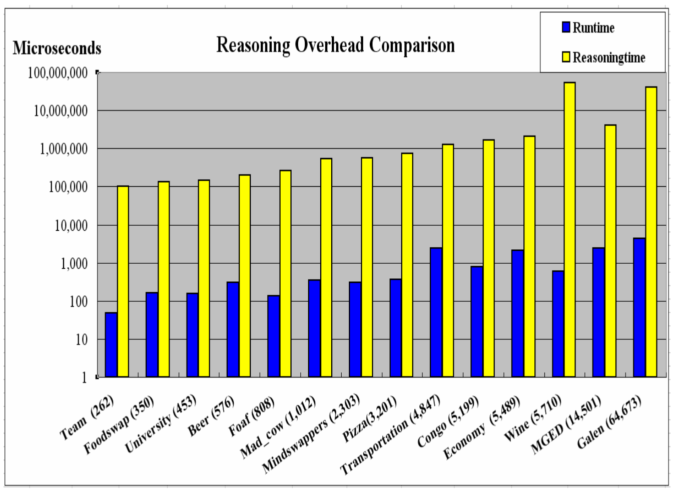

Figure 2. The effect of ontology size on reasoning time

B. Merged Ontology Reasoning

We compared the reasoning times of the merged ontologies shown in Table 2, with the combined time to reason over the two ontologies that constituted it. As shown in Table 3a, compared with the combined reasoning overhead of the two constituent individual ontologies in individual merged ontologies, we observed that most of the merged ontologies required less reasoning time. The **bolded lines** are cases where this is not the case. When the reasoning times were analysed in detail, most of the ontologies showed lower times for loading, classification (TBox), and initial class lookup. The times were very similar for consistency checking. However the realisation times (ABox) were higher for all merged ontologies. The most likely reason for this increased realisation time is that as the number of individuals increased in the merged ontologies and the reasoner spends more time searching the larger and more complex merged class hierarchical tree to find the proper class that the individuals belongs to. Another finding of this experiment is that the merged ontologies that take significantly longer to reason than their constituents have either a very large number of statements or most complex DL expressivity. For instance, the second largest and most complex merged ontology in our experiment, is the one which imported the large economy and complex wine ontologies. This also confirms our previous analysis discussed in section A on the sensitivity of reasoning performance to combination of ontology size and DL expressivity. Note from Table 2, that the DL expressivity of merged ontologies can be said to be the union of the expressivities of their constituent ontologies. Finally, we found that the degree of reduction of reasoning overhead on two similarly expressive and sized merged ontologies are comparable.

From our experience of using the KBN in different application scenarios, the occurrence of unknown concepts or properties is much rarer than the number of times that a routing ontology would be queried. Therefore we consider runtime query time to be an important aspect of KBN performance. Table 3b shows the comparison of runtime querying over merged ontologies and their constituent ontologies. Again it was found that the largest ontologies took longer to query.

TABLE 3a. COMPARING THE REASONING TIME OF A MERGED ONTOLOGY WITH THE REASONING TIME OF ITS CONSTITUENT ONTOLOGIES

Merged Ontology name	**Reasoning time: OntA** ***milliseconds : ms***	**Reasoning time: OntB** ***milliseconds: ms***	**Reasoning time: Merged** ***milliseconds: ms***
FoodUniversity	195	148	249
FoafFood	161	195	234
FoafUniversity	161	148	215
BeerMadcow	199	538	638
FoafMadcow	161	538	600
BeerMindswaper	199	555	595
FoodPizza	195	736	827
FoafPizza	161	736	890
BeerTransport	**199**	**1,265**	**1,655**
EconomyFood	2,051	195	2,263
EconomyBeer	**2,051**	**199**	**2,500**
EconomyTransport	**2,051**	**1,265**	**4,026**
EconomyCongo	2,051	1,695	3,586
EconomyWine	**2,051**	**52,379**	**90,419,878**
EconomyMGED	**2,051**	**4,122**	**9,575,935**

TABLE 3b. COMPARING THE RUNTIME QUERY TIME OF A MERGED ONTOLOGY WITH THE RUNTIME QUERY TIME OF ITS CONSTITUENT ONTOLOGIES

Merged Ontology name	**Runtime query time: OntA *microseconds: µs***	**Runtime query time: OntB *microseconds: µs***	**Runtime query time: Merged *microseconds: µs***
FoodUniversity	159	154	254
FoafFood	134	159	172
FoafUniversity	134	154	219
BeerMadcow	304	353	657
FoafMadcow	134	353	304
BeerMindswapper	304	307	456
FoodPizza	159	363	459
FoafPizza	134	363	452
BeerTransport	**304**	**2,383**	**2,803**
EconomyFood	2,158	159	1,950
EconomyBeer	2,158	304	2,205
EconomyTransport	**2,158**	**2,383**	**8,730**
EconomyCongo	2,158	793	2,340
EconomyWine	2,158	610	2,768
EconomyMGED	**2,158**	**2,380**	**8,697**

Coupling our findings from section A and section B above, we are confident that where the number of statements for each ontology is relatively low and where the DL expressivity of these ontologies are not complex we can generally predict that the reasoning overhead of a merged ontology will be bounded by the sum of the reasoning overheads of its constituent ontologies. However, a combination of very large number of statements or most complex DL expressivity breaks this prediction. We also found that we could predict the reasoning overhead of a merged ontology if we already know the reasoning overhead of a similar merged ontology.

C. *Impact of Type of Mapping Operators*

As discussed in Section V, the experiment here was designed to observe the impact that mapping operators may have on reasoning overhead. Since altering mapping operator will mainly affect only the structure of TBox classification hierarchy and ABox realisation (as opposed to loading and consistency checking), we only take classification and realisation as the measured metrics in this experiment. Recall that the three mapping operators involved were the *equivalence, sub-class, and super-class* operators, which in Figure 3 and Figure 4 are represented as EQU, SUB, and SUP respectively.

From the measurement of classification performance (Figure 3) and realisation performance (Figure 4), it is unclear whether any specific operator type has any major impact on reasoning overhead when compared to the other operators. An example from the classification timings shown in Figure 3 is the observation that reasoning with *equivalence* operator performs better than the other two operators in the FoafUniversity ontology. However, in the EconomyTransport ontology, the *equivalence* operator requires slightly more overhead than *sub-class* operator. As seen in Figure 4, three of the ten merged selected ontologies performed better with the *equivalence* operator. Yet, slightly better performance can be seen with some of the other operators in some of the merged ontologies. Overall from our analysis, we concluded that there is no direct relation between types of operators used in the mappings and reasoning overhead required to reason the resulting merged ontology. This candidate characteristic will not be considered as a factor that will influence mapping strategies selection.

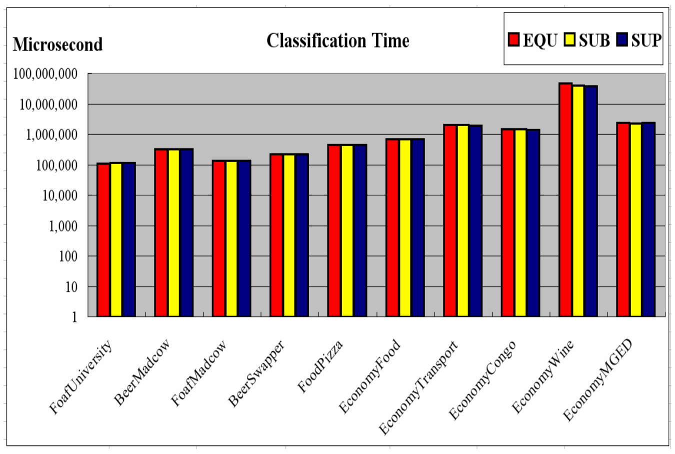

Figure 3. Classification performance for different operators

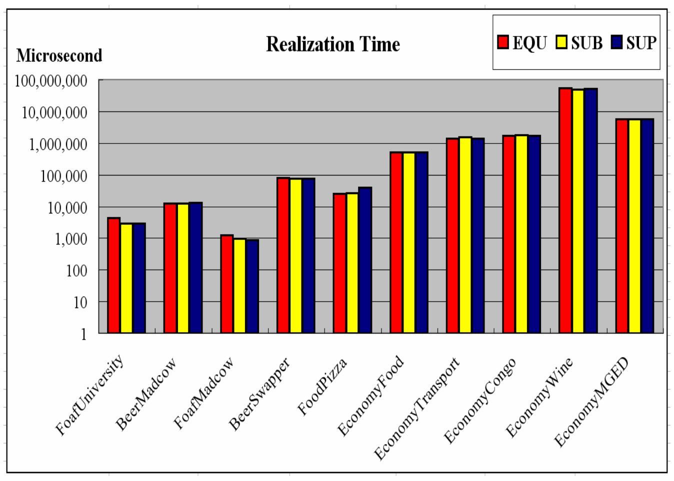

Figure 4. Realisation performance for different operators

D. *Impact of Position of Concepts used in Mapping*

As discussed in Section V, the experiment here was designed to observe the impact of mapping classes at different positions in their respective class hierarchies. Again, like the previous experiment, since selection of classes at different positions will mainly affect only the structure of TBox classification hierarchy and ABox realisation, we again only take classification and realisation as the measured metrics in this experiment. As discussed in section V we compared when the same mapping operator (*sub-class*) was applied to two top-level classes (TT), a top-level class and a bottom-level class (TB), two bottom level classes (BB) and two mid-level classes (MM). The classification and realisation performance results are presented in Figures 5 and 6 respectively.

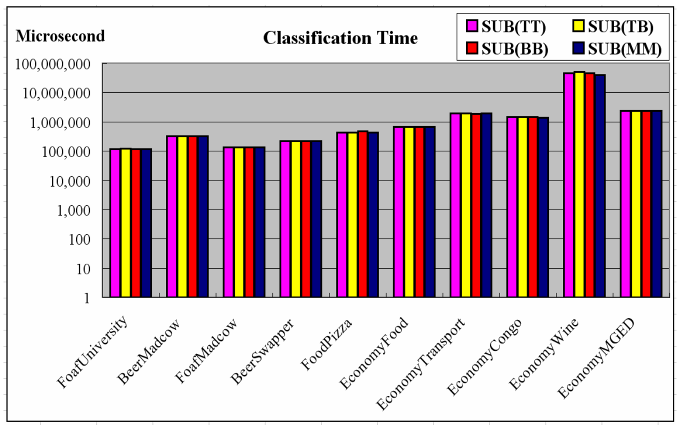

Figure 5. Classification performance for different concept positions

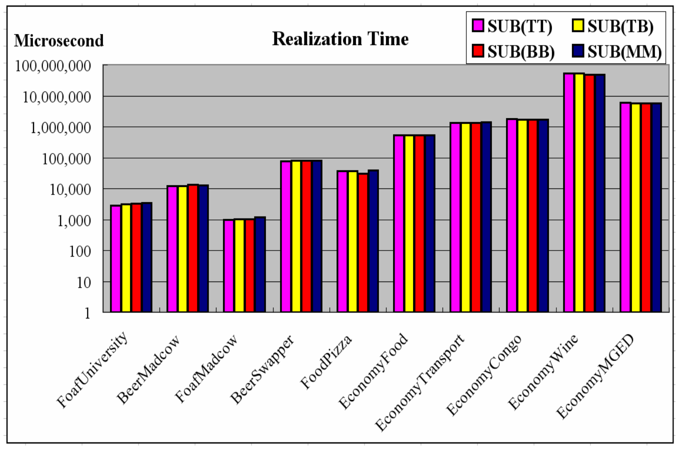

Figure 6. Realisation performance for different concept positions

As can be seen from the classification performance (Figure 5) of different class positions there is no appreciable consistent difference for the different class positions. Observing the realisation performance (Figure 6), the differing reasoning performances on different mapped classes coming from different positions is again not appreciable. For instance, the mapped classes have almost the same overhead in EconomyFood, EconomyTransport, EconomyMGED ontologies, whereas in other ontologies: some TT classes have less overhead (FoodUniversity); some BB classes lead to better performance (FoodPizza). Hence we have concluded that the concept position in class hierarchical tree is not a reliable characteristic that should be used in mapping strategies selection.

VII. Conclusion

It is very important for a contextual information routing system to be able to cope with mobile, volatile and heterogeneous set of context sources and destinations. The semantic-based publish - subscribe approach enhanced to cope with several semantic models in routing that is outlined in this paper has the potential to provide such capability. However for it to be efficient we need the loading of semantic models to be configurable depending on ontology, application and network environment characteristics. To our knowledge, the ability to configure how semantic models are to be used in routing is unique. The research undertaken for this paper has helped to clarify which ontology and mapping characteristics will be important for configuration and which are not. Namely we have concluded that the combination of size and expressivity of ontologies are the important factors for configuration. Determining which application and network environment characteristics will be important is more straightforward given previous research that has been undertaken in these areas. Thus we are now in a position to design and implement a decision component based on the characteristics of ontologies, applications and network that should configure the loading strategies so that efficient routing can be enabled.

Acknowledgment

This work is funded by Science Foundation Ireland under Grant No.03/CE3/I405: Centre for Telecommunications Value-Chain Research (CTVR) and Grant No 05/RFP/CMS014.

References

[1] Meier, R., Cahill, V., “Taxonomy of Distributed Event-Based Programming Systems“, The Computer Journal, volume 48, number 5, pp 602-626, 2005.

[2] Carzaniga, A., Rosenblum, D. S., Wolf, A. L., “The Design and Evaluation of a Wide-Area Event Notification Service”, ACM Transactions on Computer Systems, Vol.19, Issue 3, August, 2001.

[3] Segall, B. et al, “Content-Based Routing in Elvin4”, In Proceedings of AUUG2K, Canberra, 2000.

[4] Pietzuch, P., Bacon, J., "Peer-to-Peer Overlay Broker Networks in an Event-Based Middleware". in Proceedings of 2nd International Workshop on Distributed Event-Based Systems, (DEBS03), at ACM SIGMOD/PODS Conference, California, June, 2003.

[5] Chand, R., Felber, P.A., “A Scalable Protocol for Content-Based Routing in Overlay Networks”, IEEE International Symposium on Network Computing and Applications, Cambridge, MA , April 2003.

[6] Strom et al., “Gryphon: An Information Flow Based Approach to Message Brokering“, In Intl. Symp. on Software Reliability Engineering 1998.

[7] Berners-Lee, T., Hendler, J., Lassila, O., “The Semantic Web”, Scientific American, May 2001.

[8] Wang, J., Jin, B., Li, J., “An ontology-based publish/subscribe system”. In Proceedings of the 5th ACM/IFIP/USENIX international Conference on Middleware, 2004.

[9] Masuoka, R., Labrou, Yannis, Parsia, B., Sirin, E., “Ontology-Enabled Pervasive Computing Applications”, IEEE Intelligent Systems, Sep-Oct 2004, pp 68-72.

[10] Belecheanu, R., Jawaheer, G., Hoskins, A., McCann, J.A., Payne, T., “Semantic web meetings autonomic ubicomp” in Proceedings of the Workshop on Semantic Web Technology for Mobile and Ubiquitous Applications, Hiroshima, Japan, 2004.

[11] Lynch, D., Keeney, J., Lewis, D., O'Sullivan, D., "A Proactive approach to Semantically Oriented Service Discovery", in Proceedings of the Second Workshop on Innovations in Web Infrastructure (IWI 2006), Co-located with the 15th International World-Wide Web Conference, Edinburgh, Scotland. May 2006.

[12] Keeney, J., Lewis, D., O'Sullivan, D., "Benchmarking Knowledge-based Context Delivery Systems", in Proceedings of the International Conference on Autonomic and Autonomous Systems (ICAS 06), Silicon Valley, USA, July 19-21, 2006.

[13] Keeney, J., Lynch, D., Lewis, D., O’Sullivan, D., "On the Role of Ontological Semantics in Routing Contextual Knowledge in Highly Distributed Autonomic Systems",Tech. Report (TCD-CS-2006-15), Dept of Computer Science, Trinity College Dublin. 2006.

[14] Keeney, J., Lewis, D., O'Sullivan, D., "Ontological Semantics for Distributing Contextual Knowledge in Highly Distributed Autonomic Systems", Journal of Network and System Management, Special Issue on Autonomic Pervasive and Context-aware Systems, Volume 15, Number 1, March, 2007.

[15] Guo, S., Keeney, J., O'Sullivan, D., Lewis, D., "Adaptive Semantic Interoperability Strategies for Knowledge Based Networking", in Proceddings of the International Workshop on Scalable Semantic Web Knowledge Base Systems, (SSWS 2007), at OTM 2007, Vilamoura, Portugal, 21-29 Nov, 2007.

[16] Lewis, D., Keeney, J., O'Sullivan, D., Guo, S., "Towards a Managed Extensible Control Plane for Knowledge-Based Networking", in Proceedings of the 17th IFIP/IEEE International Workshop on Distributed Systems: Operations and Management Large Scale Management, (DSOM 2006), at Manweek 2006, Dublin, Ireland, 23-25 October, 2006.

[17] Petrovic, M., Burcea, I., Jacobsen, H.-A., "S-ToPSS: Semantic Toronto Publish/Subscribe System". In Proceeding of the International Conference on Very Large Databases (VLDB03), Berlin, Germany, 9 – 12 September, 2003.

[18] Cilia, M., Antollini, M., Bornhovd, C., And Buchmann, A, "Dealing with Heterogeneous Data in Pub/Sub systems: The Concept-Based Approach", in Proceedings of 3rd International Workshop on Distributed Event-Based Systems, (DEBS04), Edinburgh, Scotland, 24 – 25 May 2004.

[19] Keeney, J., Lewis, D., O'Sullivan, D., Roelens, A., Boran, A., Richardson, R., "Runtime Semantic Interoperability for Gathering Ontology-based Network Context", in Proceedings of the IEEE/IFIP Network Operations and Management Symposium (NOMS 2006), Vancouver, Canada. 3-7 April, 2006.

[20] Lefort, L., Taylor, K., Ratcliffe, D., "Towards Scalable Ontology Engineering Patterns: Lessons Learned from an Experiment based on W3C's Part-whole Guidelines ", in Proceedings of the 2nd Australasian Workshop on Advances in Ontologies, (AOW 2006), Hobart, Australia, 5 December, 2006.

[21] Pellet Performance, www.mindswap.org/2003/pellet/performance.shtml

[22] Motik, B., Sattler, U., "Practical DL Reasoning over Large Aboxes with KAON2", found at: http://www.fzi.de/KCMS/kcms_file.php?action=link&id=580, expected publication 2006.

[23] Tempich, C., Volz, R.: "Towards a Benchmark for Semantic Web Reasoners - an Analysis of the DAML Ontology Library", in Proceedings of Evaluation of Ontology-based Tools, (EON2003), at 2nd International Semantic Web Conference (ISWC 2003), Florida, USA, 20-23 October, 2003.

[24] Gardoso, J., "The Semantic Web Vision: Where are We?", IEEE Intelligent System, Sept-Oct 2007, pp.22-26, 2007.

[25] Pan, Z., "Benchmarking DL Reasoners Using Realistic Ontologies", in Proceeding of the International Workshop on OWL: Experience and Directions, (OWLED2005), Galway, Ireland. 11-12 November, 2005.

[26] Guo, Y., Heflin, J., Pan, Z., "An Evaluation of Knowledge Base Systems for Large OWL Datasets", Technical Report, CSE Dept., Leigh University, 2004.

[27] Guo, Y., Heflin, J., "LUBM: A Benchmark for OWL Knowledge Base Systems", Journal of Web Semantics, Volume 3, Issue 2, 2005.

[28] Parsia, B., Sirin, E., "Pellet: An OWL-DL Reasoner", in Proceddings of 3rd International Semantic Web Conference, (ISWC 2004), Hiroshima, Japan, 7-11 November, 2004.

[29] Carroll, J., Dickinson, I., Dollin, C., "Jena: Implementing the Semantic Web Recommendations", in Proceddings of the 13th International World Wide Web Conference, (WWW 2004), New York, 17-22 May, 2004. http://jena.sourceforge.net/

[30] Resource Description Framework: http://www.w3.org/RDF/

[31] DAML+OIL: http://www.w3.org/TR/daml+oil-reference

[32] Common Information Model v 2.10.1, DMTF 2005: http://www.dmtf.org/standards/cim/cim_schema_v2101

[33] Structure of Management Information: http://en.wikipedia.org/wiki/Structure_of_Management_Information

APPENDIX 1

Web ontologies:

Teams:
http://owl.man.ac.uk/2005/sssw/teams

Foodswap:
http://www.mindswap.org/dav/ontologies/commonsense/food/foodswap.owl

University:
http://www.mindswap.org/ontologies/debugging/university.owl

Beer:
http://www.purl.org/net/ontology/beer

Foaf:
http://xmlns.com/foaf/0.1/index.rdf

Mad_cow:
http://www.cs.man.ac.uk/~horrocks/OWL/Ontologies/mad_cows.owl

Mindswapper:
http://www.mindswap.org/2004/owl/mindswappers

Pizza:
http://www.co-ode.org/ontologies/pizza/pizza_20041007.owl

Transpotation:
http://reliant.teknowledge.com/DAML/Transportation.owl

CongoService:
http://www.daml.org/services/owl-s/1.1/CongoService.owl

Economy:
http://reliant.teknowledge.com/DAML/Economy.owl

Wine:
http://www.w3.org/2001/sw/WebOnt/guide-src/wine

MGED:
http://mged.sourceforge.net/ontologies/MGEDOntology.daml

Galen:
http://www.cs.man.ac.uk/~horrocks/OWL/Ontologies/galen.owl

Mapping ontologies:

FoodUniversity:
https://www.cs.tcd.ie/~gsong/ontologies/test/UniversityFoodMapping_Test.owl

FoafFood:
https://www.cs.tcd.ie/~gsong/ontologies/test/foaffoodMapping_Test.owl

FoafUniversity:
https://www.cs.tcd.ie/~gsong/ontologies/test/foafuniversityMapping_Test.owl

BeerMadcow:
https://www.cs.tcd.ie/~gsong/ontologies/test/beermadMapping_Test.owl

EconomyMGED:
https://www.cs.tcd.ie/~gsong/ontologies/test/EconomyMGEDMapping_Test.owl

BeerMindswap:
https://www.cs.tcd.ie/~gsong/ontologies/test/beermandswapperMapping_Test.owl

FoodPizza:
https://www.cs.tcd.ie/~gsong/ontologies/test/FoodswapPizzaMapping_Test.owl

FoafPizza:
https://www.cs.tcd.ie/~gsong/ontologies/test/foafPizzaMapping_Test.owl

BeerTransport:
https://www.cs.tcd.ie/~gsong/ontologies/test/beertransportmapping_test.owl

EconomyFood:
https://www.cs.tcd.ie/~gsong/ontologies/test/EconomyFoodMapping_Test.owl

EconomyBeer::
https://www.cs.tcd.ie/~gsong/ontologies/test/EconomyBeerMapping_Test.owl

EconomyTransport:
https://www.cs.tcd.ie/~gsong/ontologies/test/EconomyTransportationMapping_Test.owl

EconomyCongo:
https://www.cs.tcd.ie/~gsong/ontologies/test/EconomyCongoMapping_Test.owl

EconomyWine:
https://www.cs.tcd.ie/~gsong/ontologies/test/EconomyWineMapping_Test.owl