

# Adaptive Semantic Interoperability Strategies for Knowledge Based Networking

Song Guo, John Keeney, David Lewis, Declan O'Sullivan  
 Knowledge & Data Engineering Group, School of Computer Science & Statistics,  
 Trinity College, Dublin, Dublin 2, Ireland.  
 Email: {gsong, John.Keeney, Dave.Lewis, Declan.OSullivan}@cs.tcd.ie

**Abstract**— Knowledge-based networking involves the forwarding of messages across a network based on semantics of the data and associated metadata of the message content. 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. This paper examines the content heterogeneity problem in a Knowledge-based Network (KBN) implementation and evaluates a mechanism for efficiently and dynamically loading ontological mappings for use with distributed and heterogeneous knowledge-based applications. It compares a number of strategies that use pre-existing semantic mapping information stored in KBN routers. Evaluation results show that this mechanism can effectively solve the heterogeneity problem.

## I. INTRODUCTION

Given the rapid evolution and dynamism of networking, 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. Traditional Publish/Subscribe (Pub/Sub) systems [1][2] provides decoupling of identify between producers and consumers of transmitted information, but requires messages to be categorised into predefined types. In response, Content-Based Networks (CBN) have been developed [3][4][5]. These match messages to consuming client interests by specifying a filter on the messages' attribute values. Several CBN solutions and prototypes exist, e.g. [3][4][5][6][7]. 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. The limitation of current CBNs is that they only support a very limited range of datatypes and operators for use in matching consumer subscriptions to message attributes, typically: Strings, Integers, Booleans, and associated equality, greater than, less than, and regular expression matches on strings. For a CBN to work on a large scale it needs to support a richer expressiveness that can cope with the widely heterogeneous and frequently changing range of message content and consumer subscriptions.

Increasingly, researchers are turning to the use of ontology-based semantics to address this issue. Over the

years, a number of ontology languages have been developed, focusing on different aspects of ontology modelling. For example, traditional ontology language (e.g., Ontologua, F-logic) are only focusing on modelling ontology in a formal way, while the standardisation of ontology languages by the Semantic Web initiative at the World Wide Web Consortium (W3C) [8], e.g., the Web Ontology Language (OWL) [9], are more concerned with expressing and annotating metadata of information and data published by the web. The formal and decidable semantics of OWL ontologies support more expressivity than other languages such as XML and RDF, while automated reasoning allows the correctness of such ontologies to be checked automatically.

These advances have spurred an increasing number of researchers to use ontology-based semantics to support interoperability in heterogeneous and evolving systems [10][11][12]. 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 with different ontologies describing those semantics. We first provide more background on knowledge based networking and then outline a number of strategies to deal with this semantic interoperability issue. We then identify and evaluate the semantic characteristics and semantic reasoning requirements that influence the selection of one of these strategies. We also introduce the use of a Bayesian Network (BN) mechanism to dynamically and efficiently select the appropriate semantic interoperability strategy in a manner that is cognisant of the important factors that influence strategy selection. We then present a design, implementation and evaluation of KBN that implements and adaptively selects these strategies.

## II. KNOWLEDGE-BASED NETWORKING

Knowledge-based networking involves the forwarding of messages across a network based on semantics of the data and associated meta data of the message content. In previous papers [13][14][15][16] we have presented Knowledge Based Network (KBN) implementations. In a 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. The KBN broker network is then responsible for routing publication messages from producers (publishers) to interested consumers (subscribers). 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 of KBN [16] which is investigated in this paper is an extension of the Siena CBN middleware [3] to incorporate ontological datatypes and ontological subscription operators.

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 a 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.

As described in [13] and [17] the authors explore a set of applications to motivate the use of knowledge-based networking. In particular knowledge-based networking is best suited to applications where producers of semantically enhanced information are not bound to subscribers and so cannot pre-agree on subject classifications or types to label or tag messages. This approach is also ideal where subscribers require a rich and expressive subscription mechanism, above that provided by content-based networking approaches. Motivating examples of the use of knowledge-based networking, described in [13] [17], include:

- Decentralised semantic service discovery, whereby semantic web services can be composed based on expressive KBN subscriptions. [18]
- Semantic sensor readings in a multi-domain heterogeneous ubiquitous computing application [16][19]
- News distribution, whereby RSS feeds can be easily extended with semantic mark-up in Web 2.0 / Semantic Web [20]
- Semantically rich notifications from heterogeneous network elements to support multi-protocol Operational Support Systems [21]
- Distributed fault correlation, where the causal relationships between network faults can be encoded semantically rich notifications, thereby supporting automated fault correlation by way of semantic subscriptions [22]

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 building custom semantic interoperability gateways or

proxies. 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.

### III. RELATED WORK

Currently, a number of solutions utilise ontology technology in Pub/Sub systems. A comprehensive review of such systems is presented in [17], however the most influential systems are briefly described here. S-ToPSS [23] is a semantic-aware content based network, it proposed three approaches to enhance subscriptions and events semantically, in order to make the existing centralised syntactic matching algorithm semantic-aware while keeping the efficiency of current event matching technique. Another system called Ontology-based Pub/Sub system is developed by [10]. Aiming to improve expressiveness of events and subscriptions, it uses RDF and DAML+OIL techniques to describe events and subscriptions, where events and subscriptions are represented as RDF graphs and graph patterns respectively. However, this RDF graph-matching mechanism results in a system that can no longer perform simple (non-semantic) subscription matching. Another system described in [24] embeds ontological topic information with tagged/named values in XML-based messages. Subscribers can then subscribe to semantic topics using the SPARQL query language. Here all processing is performed at the client node, and again like the previous system does not support simpler non-semantic subscriptions. Reference [25] presents an independent concept-based layer which is built between the notification service and the Pub/Sub application to provide a high level interaction among applications, in order to tackle the problem of event interaction among heterogeneous applications. There the semantic extension is performed at the edge of the network rather than deep in the network as demonstrated in this paper. Furthermore, our previous work [21] demonstrated how through the use of ontology and ontology mapping techniques applications built according to different standards (CIM and SMI were used) could interchange fault alarms over a Content Based Network (Elvin) [4] using an ontology based approach. Again, this system added semantics only at the edge of the network. 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.

### IV. SEMANTIC MAPPINGS

A Semantic Mapping is defined as the establishment of correspondences between a set of source ontologies. In our work we assume that the ontologies are expressed in the web ontology language (OWL) [9]. There is still no common way to specify the mappings using one particular language. In this work OWL itself is also used

to describe the mappings of the ontologies. OWL’s standardisation and direct support for semantic relations (i.e., equivalence, subclassof etc) made it the ideal choice for use here. In particular, with OWL, we use *equivalence*, *subsumes* and *subsumed by* relationships to express the mappings. (The *subsumes* relationship describes the super-class and super-property relationships. *Subsumed by* captures the sub-class and sub-property relationships. *Equivalence* can be used with classes, properties and individuals.) If more complex mapping types are required, another mapping language might be chosen, but the choice of mapping languages must be considered carefully in order to maximise interoperability and mapping reusability.

In order to illustrate this approach we give a concrete mapping example, however, the approaches discussed in this paper are independent of the actual contents of the ontologies. Fig. 1 describes a selection of mapping relations between classes and properties from two ontologies. The ontologies *xgpl-regn1* and *xgpl-regn2* are region description ontologies, made by the authors. To make a mapping between these two ontologies we first need to import both into a new mapping ontology so that the rest of ontology description will be able to refer to the existing elements that are previously defined in an involved ontology. Second, we establish mappings between elements of the involved ontologies. For instance, one class of an ontology may be considered as a subclass of another class of another ontology (*xgpl-regn1:Village* is a *subclass* of *xgpl-regn2:RuralUnit* in Fig. 1). Finally, two relations (*subsumption* and *equivalence*) between properties from the involved ontologies can be determined by comparing their members (*xgpl-regn1:ishouseof* is a *subProperty* of *xgpl-regn2:isbuildingof* in Fig. 1).

Now, let us assume that *xgpl-regn1* is the main application and routing ontology distributed among some KBN routers. If there is a service provider interested in event about the *xgpl-regn1:City* concept, it subscribes a query expressed by concept *xgpl-regn1:City* to its closest KBN router. If this router receives a notification that has the same attribute name as used in the subscription filter, with a value that is an ontological concept, but the concept is *xgpl-regn2:EuroCity* (“*EuroCity*” is not defined in *xgpl-regn1* but rather in *xgpl-regn2*), this KBN

router needs to explore the mappings to find mapping relations containing “*EuroCity*” and “*City*” to resolve this unknown concept. In this case, the mapping ontology is explored, where the concept “*City*” is identified as superclass of the concept “*EuroCity*”.

V. SEMANTIC MAPPINGS IN A KBN ROUTER

Before discussing how our KBN router model was extended to support semantic interoperability, the original KBN router model must be briefly discussed. Our KBN router is an extension of the Siena content-based router [3]. A Siena notification is a set of typed attributes. Each attribute is comprised of a name, a type and a value. Siena supports the following attribute types: string, long, integer, double and boolean. A Siena subscription is a conjunction of filtering attribute constraints. A constraint is comprised of the attribute name, a comparison operator, and a value. A subscription covers a notification if the event satisfies all filtering constraints of a filter. A notification is delivered to a client if the client has submitted a subscription filter satisfied by that notification. Siena also discovers aggregations between filters to optimise the subscription tree (subtree) at each router. As new subscriptions arrive at a router the subscription tree is searched to find the appropriate position to insert the new subscription.

As mentioned the KBN implementation upon which this work is based extends Siena. In particular it was extended with three new ontological datatypes: Class / Concept, Individual / Instance, and Property; and three new ontological subscription operators: More-specific (sub-class / sub-property), Less-specific (super-class, super-property), and Equivalent (equivalent-class, equivalent-property, same-as-individual). To achieve this, each KBN router holds a copy of an ontology, within which each ontological class, property and individual is described. A more detailed discussion of the KBN router model, and how it is extended from the Siena CBN router, is presented in [15].

The KBN router design was then *further* extended to support heterogeneous ontologies in the network. Each extended KBN router is implemented with two ontology repositories: the main application ontology store provides the ontology for the KBN operation, whereas the mappings in the mapping ontology store are used for

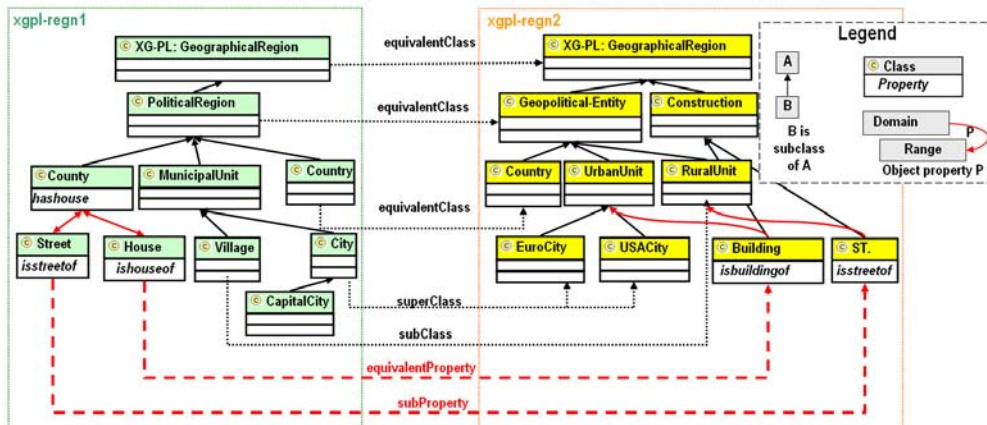


Figure 1: Semantic Mappings

helping the KBN router achieve semantic interoperability. In the original KBN router, every router had a copy of the same main application ontology, however in this extension each router can have a different local main application ontology, and a different set of mapping ontologies to support interoperability between application ontologies. All ontologies and mappings are provided by the administrators of the network. The ontology registration interface allows administrators to register both application ontologies and mappings with KBN routers. Both publishers and subscribers register with KBN router via a client registration interface, and they need to provide their own ontology that defines the knowledge bases used by the clients in their subscriptions and notifications.

Subscriptions can arrive at the KBN router either directly from a client or forwarded from another node in the KBN network. The query subscription, using terms from the subscriber’s local ontology, is passed to the subscription tree (subtree) searching engine, which searches the subtree and inserts the subscription in the appropriate position. However, the subscription may use ontological terms that are not contained in the router’s local ontology, so the position to insert the subscription into the subtree cannot be immediately resolved.

Similarly, when a publication arrives at a KBN router, either directly from a client or from another KBN node, the subtree searching engine walks the subtree to find appropriate matching subscriptions to find the set of subscribers (clients and other KBN nodes) that should be notified with the publication. Again, the publication may use ontological terms that are not contained in the router’s local ontology, so the set of matching subscriptions cannot be immediately resolved.

If the subtree searching engine receives a subscription or publication with ontological terms which are not expressed by terms from the application ontology, the mapping management interface is called to explore the mapping store where the mappings were previously injected (see Fig. 2).

VI. SEMANTIC INTEROPERABILITY STRATEGIES

If subscriptions or publications contain heterogeneous semantic content then an individual KBN router will occasionally encounter an unknown concept (or individual or property) that is not described in its own routing ontology. When a KBN router (Fig. 2) 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 perform this searching and merging of mappings in an efficient manner. Currently there are three strategies available to incorporate semantic mapping information into the KBN router’s routing ontology as follows:

- The “Every mapping file” Strategy (*Every*): forces the router to load all available mappings and imported ontologies (referenced in the mappings) into its

routing ontology at once. This strategy maximises the exploration of mappings to tackle the unknown data problem.

- The “Appropriate mapping file” Strategy (*ApproOnly*): the KBN router checks available mapping files for mappings that contain at least one concept used by the conflicting subscription or notification. It then merges the appropriate mappings and their referenced (imported) ontologies.
- The “Appropriate individual mapping” Strategy (*ApproInd*): 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. It does not include any referenced imported ontologies, just the individual mapping.

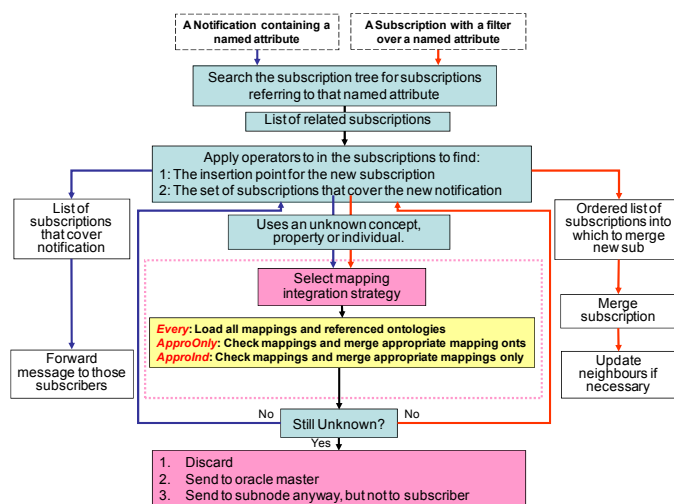


Figure 2: Semantic Interoperability in a KBN router

VII. FACTORS AFFECTING 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, matching and routing performance of a KBN broker 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 a number of factors, come of which may be

changing dynamically. We categorised the possible trigger factors into three main sets: characteristics of the ontologies; application characteristics; and environmental and networking characteristics. Given different possible mapping strategies, our recent research has focused on identifying which of the ontology, application and environmental characteristics 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.

#### A. Ontology Characteristics

Firstly, ontology characteristics that may impact strategy selection at an individual router are: the current size of the routing ontology, the size of the mapping ontologies, the current complexity (DL expressivity) of the routing ontology, complexity (DL expressivity) of the mapping ontologies and their associated referenced imported ontologies. This set of factors was chosen from a large set of possible factors, as described in a previous publication [19], which concluded that this combination of number, size and expressivity of the ontologies are the most important ontology characteristics to influence the strategy selection component. These factors are important as indicators of the potential memory, time and processing overhead involved in merging and reasoning mappings and referenced ontologies into the broker's own routing ontology, and the later overhead of querying these ontologies once loaded [26].

#### B. Application Characteristics

Secondly, the application characteristics that may impact strategy selection at an individual router are: the rate of publications and subscriptions arriving at the broker; the tolerance capability of KBN applications to respond gracefully to a false-positive/-negative match between a subscriptions and publication that may be due to a mapping being missed; the observed occurrence rates of unknown (not expressed by the routing ontology) ontological classes (and properties and individuals) in publications or subscriptions. Since the KBN network is essentially a middleware, built to facilitate the applications that operate above it, these factors are particularly important in allowing some of the characteristics of the application to inform the semantic interoperability strategy of the brokers/routers in the KBN network.

#### C. Environmental Characteristics

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 at the individual router; and the number of mapping ontologies stored in the KBN router, and the number of ontologies

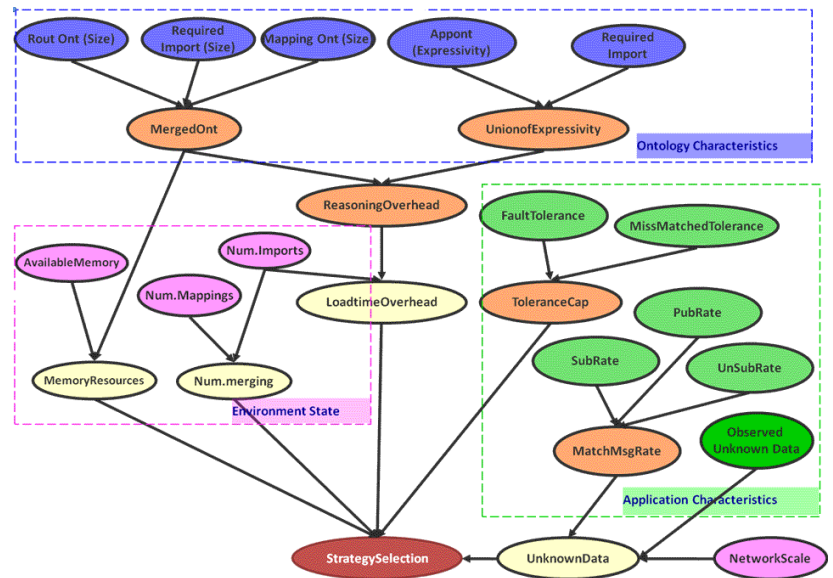


Figure 3: How the factors affecting strategy selection influence each other

imported or referenced by the mappings. It is important to note that each router/broker independently determines the appropriate strategy to use, yet individual (and possibly heterogeneous) brokers operate together as part of a cooperating network of brokers to achieve a decentralised knowledge distribution network. Therefore these network and environmental factors are required so brokers are cognisant of their resources and the topology in which they operate.

### VIII. A BAYESIAN NETWORK TO SELECT THE APPROPRIATE STRATEGY

Table 1 summarises the possible combinations of states of 11 of the most influential aggregated trigger factors that have significant impact on selecting mapping strategies. Therefore, based on this table, the human administrator could select the proper mapping strategy for an individual router through the observation of the combinations of the states of trigger factors. However, selecting mapping strategies purely based on Table 1 proves to be a complicated and difficult task due to the following reasons:

- As stated there are 15 individual trigger factors identified and each factor can have two or three states resulting in up to  $3^{15}$  combinations of factors.
- Not all of the trigger factors have a direct independent impact on the decision making process, instead, some of the factors are interdependent and cumulative. For example, the network scale, message rate and observed unknown data rate together determine the rate of unknown data occurrences, which then has significant impact on selecting mapping strategies.
- Table 1 is overly simplistic since there exists no determining factor to drive strategy selection. Instead, a mechanism to weigh the benefits and deficiencies of each strategy for a given observation of the factors is required.

Table 1: Which strategy is appropriate for some factors

	Application Trigger Factors				Environment Trigger Factors			Ontology Trigger Factors			
Strategy	Pub, sub unsub rate/active/inactive/cycle duration	End user tolerance (fault occurrences)	End user tolerance/missed matches	Observed Unknown data Rate	Network Scale	Memory resources allocated in Broker	Number of Mapping and reference ontologies	Size of an individual mapping ontology (required)	Size of routing ontology of the broker	Complexity of routing ontology (DL Expressivity)	Complexity of mapping ontology (DL Expressivity)
Every	Large	Small	Small	Large	Small/Medium/Large	Large	Small	Large	Large	Large	Large
ApproOnly	Medium	Medium	Medium	Medium	Small/Medium/Large	Medium	Medium	Medium	Medium	Medium	Medium
ApproInd	Small	Small	Small	Small	Small/Medium/Large	Small	Large	Small	Small	Small	Small

- In a large scale deployment, as the factors influencing strategy selection change dynamically at each broker, it is infeasible to require a human administrator to manually select the appropriate strategy.

It is worthwhile to explicitly express the relations between trigger factors and the selection/prediction of mapping strategies as shown in Fig. 3 as an influence diagram [27], a compact graphical representation of a decision situation for linking trigger factors and mapping strategies selection. The leaf nodes represent the 15 application-, environment-, and semantic-level identified strategy selection factors. Intermediate/internal nodes represent combinations and interdependencies in the factors, while the root node represents the combination of factors to influence strategy selection.

While the influence diagram in Fig. 3 shows the dependency grouping between factors it does not show the degree to which individual factors, and combinations of factors, should influence strategy selection. For this a probabilistic/weighting-based approach is required. A Bayesian Belief Network [28] was chosen as the most appropriate approach to achieve this. A Bayesian Network represents the probabilistic relationships between variables and allow predictions based on expert knowledge, historical data, and causal/correlation relationships. Each variable, represented as a node in an acyclic graph, can have a number of states and weighting for each state representing the probability that the variable will have that state/value. The state value for leaf variables is given or observed, and the weightings for each state of the internal variables is derived from a weighted combination of its input child variables' states. This can proceed through a number of levels until a root variable's state/value weightings represent the aggregation of all of the variables' states in the graph.

A Bayesian Belief Network was constructed based on the causality/influence hierarchy, shown in Fig. 4. The

leaf nodes represent the observed variables, and the root node's possible states represent the selection of one of the three semantic interoperability strategies. Therefore, the root node's output state with the highest probability/weighting is therefore the appropriate strategy to use.

In order to develop a credible and validated model for modelling mapping strategy selection decision-making, the author followed the guidelines proposed by [29][30] to develop, test and revise our Bayesian Network (BN) model to avoid potentially spurious or unreliable probability models. First of all, we developed an initial parameterised BN model. We then iteratively tested the BN model by using various combinations of input values and sensitivity analysis to readjust the network structure and state occurrence probabilities/weighting until it responded reasonably. We then tested the BN model with a series of case data to test the accuracy of model for the purpose of handling missing data on some nodes or states, after which the eventual model is produced, as shown in Fig. 4. Due to space constraints the rigorous and time-consuming process of finalising the belief network and the factor weighting are omitted from this paper.

The final model, as shown in Fig. 4 consists of the 14 leaf variables or inputs (measured or observed), 7 internal nodes whose derived state probabilities are based on weighted combinations of their children's state values, and a single root node whose weightings indicate which strategy is most appropriate for a given set of values for the input variables. It should be noted that although the authors have painstakingly calculated default weightings, rankings, influence paths, and boundary values for each of the variables, these configurations are all dynamically customisable for different KBN deployments. (Note, in Fig. 4, the occurrence probabilities for each of the states at the leaf nodes are equal and so insignificant since the given state for these nodes is actually observed or given rather than calculated so the weightings are ignored).

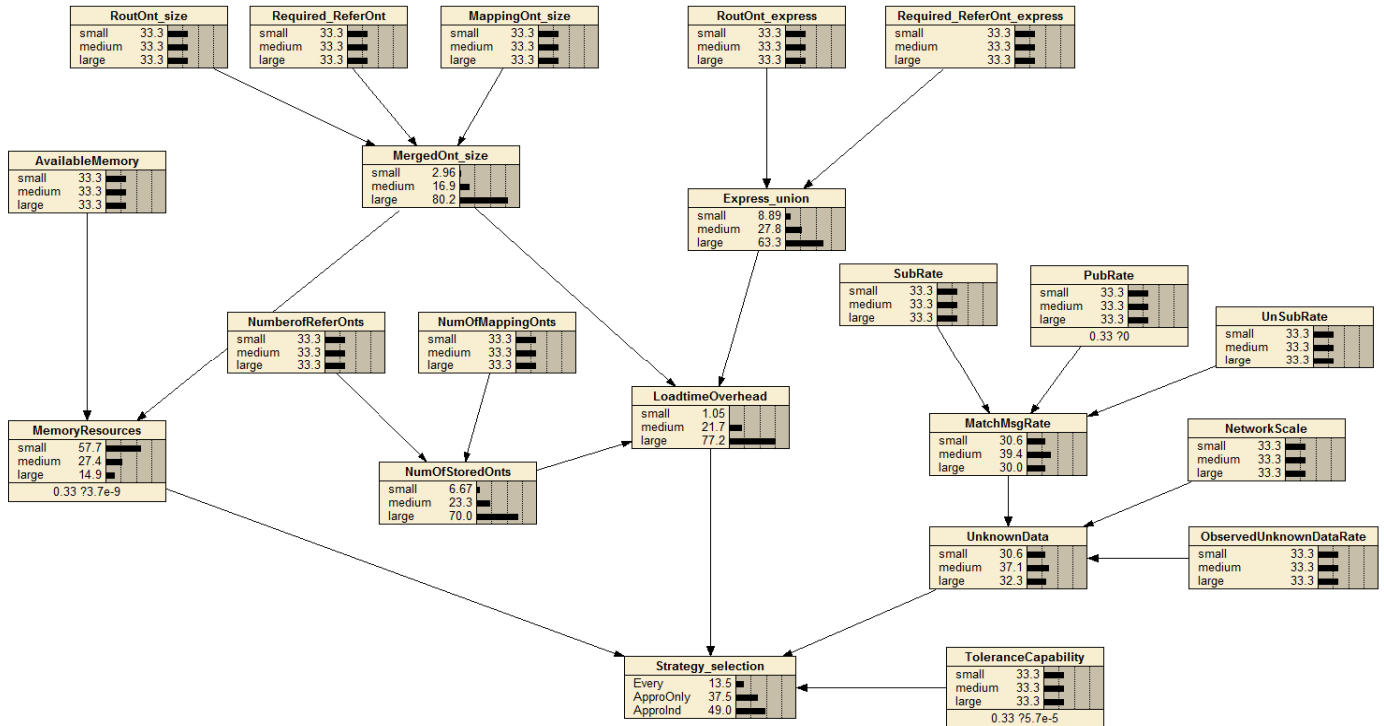


Figure 4: The Bayesian Belief Network for selecting the appropriate semantic interoperability strategy

A. Ontology-based Nodes

“RouteOnt\_size” refers to the current size, measured as the number of statements, of the current routing ontology. This is input variable is automatically measured and maintained by the KBN router. Ontology models with less than 1000 statements are regarded as small, 1000 to 5000 statements are medium, and models with greater than 5000 are regarded as large. These boundary values are based on previous experiments by the authors [19][26].

“MappingOnt\_size” refers to the size of mapping ontologies stored at the router, again measured as the number of statements, ranging in size from small (<1000), medium (≥1000, <5000), and large (≥5000).

“Required\_ReferOnt” refers to the size of the ontologies referenced by the mapping ontologies, again measured as the number of statements, ranging in size from small (<1000), medium (≥1000, <5000), and large (≥5000).

“MergedOnt\_size” is the probability that a newly merged routing ontology incorporating mappings will be small (<1000), medium (≥1000, <5000), or large (≥5000). This is conditional on the “RouteOnt\_size”, “Required\_ReferOnt”, “MappingOnt\_size” nodes.

“RouteOnt\_express” refers to the reasoning complexity or DL expressiveness<sup>1</sup> of the routing ontology.

<sup>1</sup> 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, i.e. *ALC*); *R* - Role Transitivity; *H* - Role Hierarchy; *I* - Role Inverse; *O* - Nominal; *N* - unqualified number restrictions; *Q* - qualified number restrictions; *F* - only functional number restrictions; *SH* denote the extension of *ALC* with transitive and hierarchical roles (*ALC<sub>HR+</sub>*); (*D*) - Datatypes.

Ontologies with an expressivity of *ALC* or less are regarded a low expressivity, with ontologies ranging from *ALC* up to *SHOIN* being regarded as medium expressivity, with *SHOIN* and *SHOIQ* ontologies being considered highly expressive. These boundary values are based on previous experiments by the authors [19][26]. Like size, the expressivity of an ontology can be found by loading the ontology and then querying the ontology reasoner, however, with the aim of minimising the loading of ontologies these values can also be found from lookup tables.

“Required\_ReferOnt\_express” refers to the DL expressivity of the ontologies referenced by the mappings, again ranging between low (<*ALC*), medium (≥*ALC*, <*SHOIN*), and high (*SHOIN* & *SHOIQ*).

“Express\_union” is the probability that the expressivity of a newly merged routing ontology incorporating mappings will be low (<*ALC*), medium (≥*ALC*, <*SHOIN*), and high (*SHOIN* & *SHOIQ*). This is conditional on the “Required\_ReferOnt\_express” and “RouteOnt\_express” nodes. The DL expressivity of a merged ontology is the union of the expressivity of its constituent ontologies.

B. Environment-based Nodes

“NumerofMappingOnt” refers to the number of mapping ontologies stored at a router. Less than 5 mapping ontologies is considered small, 5 to 10 mapping ontologies is medium, with more than 10 being considered high. These boundary values are based on the authors’ empirical analysis.

“NumerofReferOnt” refers to the number of ontologies referenced by the mapping ontologies, again ranging between small (<5), medium (≥5, <10) and large (≥10).

“NumOfStoredOnt” is the number of ontologies available on a KBN router, ranging between small ( $<5$ ), medium ( $\geq 5, <10$ ) and large ( $\geq 10$ ). It is derived from the “NumberOfMappingOnt” and “NumberOfReferOnt” nodes.

“AvailableMemory” refers to the memory allocation given to the KBN router application. An allocation of 64MB or less is considered small, 64MB to 120MB is considered medium, with greater than 120MB being considered high. These boundary values are based on our knowledge of the implementation of the KBN router application, our experience with different reasoning and load configurations, and the availability of memory resources on mid-range developer workstations. The value for available memory can be detected from the runtime environment within which the KBN router application runs

“MemoryResources” is the probability that the amount of memory resources available to the KBN router once mappings are loaded will be small ( $<64\text{MB}$ ), medium ( $\geq 64\text{MB}, <120\text{MB}$ ), or large ( $\geq 120\text{MB}$ ). This is conditional on the “AvailableMemory”, “MergedOnt\_size” nodes.

“LoadtimeOverhead” is the probability that the amount of loadtime initialisation and reasoning of a merged ontology will be low, medium or high. This is conditional on the “MemoryResources”, “Express\_union”, “NumOfStoredOnt” nodes. The weightings and rankings of input nodes to produce probability weighting for this node is derived from our previous work in determining the main factors that influence reasoning overhead [19][26][31]

### C. Application-based Nodes

“PubRate” refers to the average rate at which publications arrive at router/broker. A rate of 10 or less publications per minute is considered low, 10 to 20 is considered medium, while more than 20 is considered high. This is a subjective ranging of boundary values, cognisant of the authors’ observations of normal operation and loading capability of a KBN router, especially when combined with normal simultaneous loadings from subscription and unsubscription requests. The observed input value for this variable is calculated and maintained automatically by the KBN router application.

“SubRate” refers to the average rate at which subscription requests arrive at the router, ranging between low ( $<10/\text{min}$ ), medium ( $\geq 10/\text{min}, <20/\text{min}$ ) and high ( $\geq 20/\text{min}$ ).

“UnsubRate” refers to the average rate at which unsubscription requests arrive at the router, ranging between low ( $<10/\text{min}$ ), medium ( $\geq 10/\text{min}, <20/\text{min}$ ) and high ( $\geq 20/\text{min}$ ).

“MatchMsgRate” is the probability that the rate of publication to subscription matching rates will be low, medium or high. This is conditional on the “PubRate”, “SubRate”, and “UnsubRate” nodes.

“ObservedUnknownDataRate” refers to the average rate per minute at which an unrecognised ontological concept (or individual, or property) is detected at a KBN

broker. Where unknown data can occur in a publication, subscription or unsubscription we estimate that 3 occurrences per minute is low, 3-7/minute is medium, and more than 7/minute is high.

“NetworkScale” refers to the number of brokers comprising a KBN network. It is impossible to define concrete boundary values for this variable since it is inherently subjective. We estimate that a deployment with less than 5 brokers is small, 5 to 15 brokers is medium, and 15 or more is large. However, since this refers to just the number of brokers, not the number of clients attached to any broker, 15 brokers have the potential to support several thousand clients in a scalable manner.

“UnknownData” is the probability that the aggregate rate of unknown data occurrences will be low, medium or high. This is conditional on the “MatchMsgRate”, “NetworkScale”, and “ObservedUnknownDataRate” nodes.

“ToleranceCapability” refers to the tolerance the KBN clients may have when dealing with false-positive or false-negative subscription matches due to potentially missed semantic relationships due to the overly conservative loading of mappings. This variable is purely subjective, and we have not attempted to define boundary values to the low, medium and high states. We delegate the encoding of this variable to an administrator. It is envisioned that all brokers in a KBN might have the same value for this variable.

### D. Final Strategy Selection Node

The “Strategy\_selection” node then provides a set of weightings ranking the three semantic interoperability strategies, thereby deciding the appropriate mapping strategy. This conditional on available memory resources (“MemoryResources”), ontology loadtime initialisation overhead (“LoadtimeOverhead”), mismatch tolerance (“ToleranceCapability”), and the unknown data occurrence rate at the router (“UnknownData”), each of which continuously change. The rankings between inputs and the probability weighting for this node is also based on our previous work [19][26][31].

## IX. EVALUATING THE AUTOMATIC SELECTION OF KBN SEMANTIC INTEROPERABILITY STRATEGIES

As described above, based on a KBN implementation [15] derived from the Siena CBN system, an extension to support different semantic interoperability strategies was designed and built. This was then further extended into a system called *KBNMap* to support the dynamic and adaptive selection of appropriate semantic interoperability strategies as suggested by a Bayesian Belief Network presented in the previous section. The adaptive semantic mapping service provides a mechanism for the KBN brokers to self-select the mapping strategies at runtime to adapt to the changes of ontology, application and environment relevant key factors. This section presents a subset of the evaluation experiments carried out to verify the correct operation of KBNMap. The adaptability of the mapping service was also



considered in order to evaluate KBNMap's support for adaptive semantic interoperability through the use of ontology mappings and probabilistic modelling techniques. The second evaluation goal was to assess the applicability of this mechanism for adaptively selecting mapping strategies, by evaluating the correct strategy weightings inferred and calculated from the probabilistic combination of factor weightings, given different combinations of observed values/inputs/evidence for key factors.

However, since the Bayesian Network has 14 input variables, most with 3 possible states each, investigating the performance of KBNmap under such huge number combinations seems to be impractical. Indeed should such a scheme be applied to other semantic pub/sub systems then there may exist even more influential factors, with different numbers of states for each. Therefore, it is worthwhile to identify the key factors having the most significant impact on selecting mapping strategies, so that the KBNmap routers could be evaluated far fewer combinations of varying evidence. In addition, as KBNMap is mainly targeting at resolving heterogeneous information, the evidence of selected trigger factors need to be grouped into sets on the basis of combining the unknown data factor with other important factors. The method in evaluating the introduced mechanisms was divided into a number of stages. The first stage was based on our previous work [19][26][31] to define and categorise the set of the most important trigger factors to build a smaller set of test-cases. The next stage was to provide a routing ontology, a number of mapping files, and referenced ontologies with varying ontology characteristics for use by the applications and routers. The following stage was to design the general network setup, which includes a hierarchical topology of brokers and a mechanism to synthetically generate a configurable test-set of publications and subscriptions to inject into the KBNMap deployment. The final stage was to define a set of measurable performance key metrics.

The selection of the most important factors was in accordance with the ranking of direct parent nodes of BN model described in the previous section. The selected factors (in order of importance) are:

- "Observed unknown data rate" this factor is selected as the one that has the most significant influence on selecting mapping strategies since it mainly determines the rate of unknown data that actually received by the broker. For example the "Every mapping file" Strategy (*Every*) is well-suited for the environment where the unknown data is large, while the "Appropriate individual mapping" Strategy (*ApproInd*) is best where the unknown data occurrence is small. The "Appropriate mapping file" Strategy (*ApproOnly*) is a moderate trade-off in between.
- "Subscription and Publication rate": these two factors are summarised as the message rate. The rate of messages determines the rate of message matching executed by the KBN broker where the unknown data are identified. When taken together

with the "unknown data rate" the occurrence rate of subscriptions and publications with ontological terms not expressed by the routing ontology can be determined.

- "Tolerance capability": this factor is selected since a low tolerance of missed mappings, resulting in false-positive or false-negative subscription matches, is a strong indicator that the *ApproOnly* and especially the *ApproInd* strategies may be inappropriate unless other factors outweigh this factor's influence.
- "Available memory resources" is the last major influencing factor due to the differing resource consumptions of the mapping strategies. The *Every* strategy is preferable when the available resource are sufficiently large, while the *ApproInd* strategy is best-suited when memory is limited since only very targeted individual mappings are loaded.

Of the four main factors, the "Observed unknown data rate" factor is the most influential, so we created 3 test cases where each of the other 3 factors were considered in terms of their effect in combination with the "Observed unknown data rate" factor.

For the ontologies, it was decided to choose a set of real-world ontologies rather than creating ontologies ourselves. Our motivation for this is that they should be widely used data sets for evaluating current ontology matching systems, of relatively high quality, that they should be created by different people with diverse technical backgrounds and that the ontologies should range from small to large and from simple expressivity to complex expressivity. For this reason, we selected existing ontologies and their existing mapping files from Conference Track in the OAEI website [32], which is a well-known coordinated international contest to evaluate state of the art ontology matching systems. This test set introduces fourteen ontologies collected to benchmark semantic interoperability tools. In addition to the ontologies the dataset also contained a large set of semantic mappings between the ontologies. These mappings were then transformed into a format suitable for use by KBNMap and divided into smaller sets of ontological mapping ontologies. Since many of the mappings used the *ekaw* ontology (from [32]) as their canonical ontology, this ontology was chosen as the main routing ontology.

As a research goal of this work is to distribute heterogeneous information in a large-scale networking environment, the topology adopted in this evaluation concentrates on a simple large hierarchical topology, which consists of 15 nodes (large-scale) running as dedicated KBN routing brokers. This hierarchical topology, shown in Fig. 5 arranged the brokers as deep as possible, with a publishing client on one side and a subscribing client on the other side, thereby artificially simulating a long route between clients to stress the KBNMap deployment. Each broker was pre-loaded with the *ekaw* ontology as their routing ontology and the large set of mappings and referenced ontologies pre-loaded into their mapping stores. Both of the clients were pre-loaded

with the *ekaw* ontology, and 13 other ontologies that were mapped to the *ekaw* ontology. The clients then inject synthesised semantic publications, subscriptions and unsubscriptions drawn from these 14 ontologies into the network.

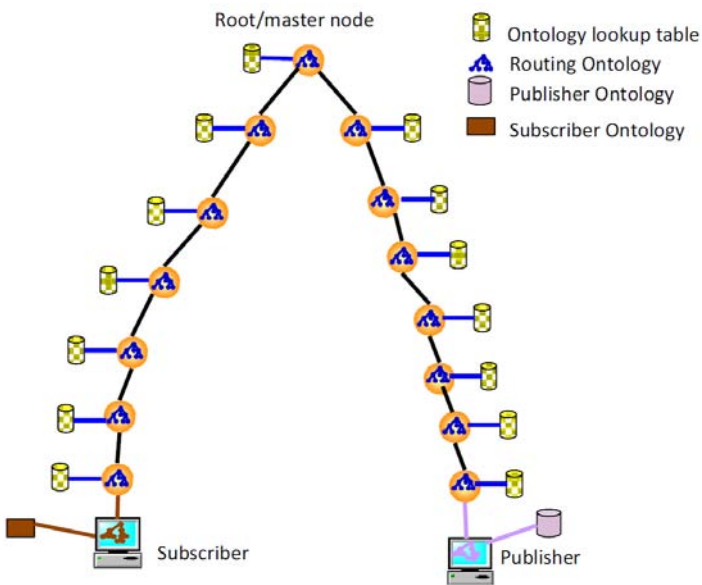


Figure 5: An Experimental Hierarchical KBN topology for evaluation

A. Comparing the Strategies

Prior to evaluating the effect of introduced adaptive mapping service on the performance of KBNMap routers it is worthwhile to comprehensively understand the extra overhead introduced by each particular individual mapping strategy. KBNMap was first initialised to employ statically encoded (hard-wired) strategy selection for each of the 3 strategies. We then tested the performance of each individual strategy with respect to different levels of unknown data. 1) where the rate of messages across the network is high while the proportion of unknown data contained in messages is relatively small; 2) where the rate of messages across the network is high while the proportion of unknown data contained in messages is raised to a medium level; and 3) where the rate of messages across the network is high while the proportion of unknown data contained in messages is raised to a high level. Table 2 represents the configuration of the parameters in detail. For the purpose of evaluating networks in high load configurations, the message rates are specified as large (30 messages/min), while the observed unknown data rate ranges from small (2 messages/min) to large (12 messages/min) in different cases. In addition, the memory resources allocated to each router (40 Mb) and tolerance capability of end applications are specified as small, so each mapping strategy works in a worst-case deployment configuration.

Table 2: Factor configurations to compare semantic interoperability strategies

	Variable Name				
	PubRate (/min)	SubRate (/min)	Unknown Data Rate (/min)	Memory (MB)	Tolerance
small unknown data	30	30	2	40	small
medium unknown data	30	30	5	40	small
large unknown data	30	30	12	40	small

The first graph of Fig. 6, which shows the time taken to execute each strategy *once*, confirms our observations the *ApproInd* strategy is efficient for a single execution and is well-suited for the situation where the unknown data is rare, while the *Every* strategy is well suited for the situation where the unknown data occurrence is very large since its high “once-off” cost enables all potential unknown data to become known in one operation. Furthermore, considering the combination of characteristics and performance of the *ApproOnly* strategy it is well-suited for the situation where the unknown data occurs at intermediate rate.

The second graph in Fig. 6, shows the *sum* of all strategy execution times across the entire network over a period of just over 1 hour. Here the *Every* strategy runs just once on each broker with a high once-off cost on each, while the other strategies must run repeatedly but with relatively low cost for each execution.

These findings only discuss the impact of the high or low occurrence of unknown data on the selection of an appropriate semantic interoperability strategy. It should

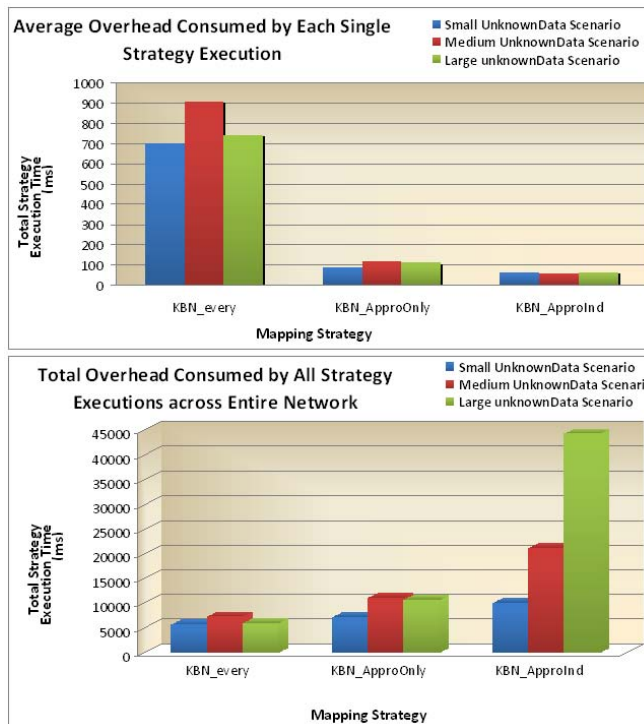


Figure 6: Comparing the semantic interoperability strategies

be noted that the other factors, such as network size have a major effect on strategy selection, thereby affirming our assertion that a more intelligent mechanism, such as our Bayesian Belief Network, is required for appropriate strategy selection.

*B. Case 1: High Unknown Data Rate with High Message Rate*

In this experiment we evaluate the combination of two factors: the message arrival rate and the unknown data rate. While all combinations of these factors were tested, here we present a scenario where the unknown data rate was high (12 occurrences/minute) and the message rate was also high (publication rate and subscription rate were both set to 30/minute). All other factors were set to default high-load settings, for example, the large network deployment/topology shown in Fig. 5, memory resources were restricted to low (40MB), mismatch tolerance was set to low, with a large set of mappings and referenced ontologies available at each router. With these inputs the Bayesian Network produced a weighting of 42% for the *Every* strategy, 57% for the *ApproOnly* strategy, and 1% for the *ApproInd* strategy. The adaptive KBNMap implementation, incorporating a dynamically updated Bayesian Network, was then compared to an implementation where the semantic interoperability strategy was selected randomly from the set of 3 strategies (KBNRan).

As the experiment ran for over 1 hour, once for each KBN implementation, it is clear from Fig. 7 that the KBNRan deployment incorrectly executed the *Every* strategy early on, resulting in very high end-to-end publication matching times. However, once the mappings were loaded in most of the brokers the KBNRan deployment performed well. In contrast, the KBNMap deployment chose the *ApproOnly* strategy which resulted in lower publication matching times. In addition, at approx. 800 seconds into the experiment, most of the KBNMap routers adaptively switched to the *ApproInd* strategy. This was due to diminishing memory resources as the routers continuously loaded mappings and merged referenced ontologies, where the *ApproInd* strategy consumes less memory.

For this experiment we also measured the end-to-end delay of the subset of the publications that matched a subscription and so were routed across the entire network over 15 hops. From Fig. 8 it is clear that KBNMAP performed better than KBNRan.

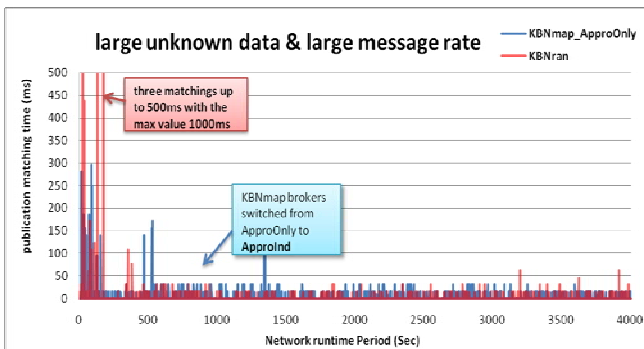


Figure 7: Pub Matching Time: High Unknown Data Rate with High Message Rate

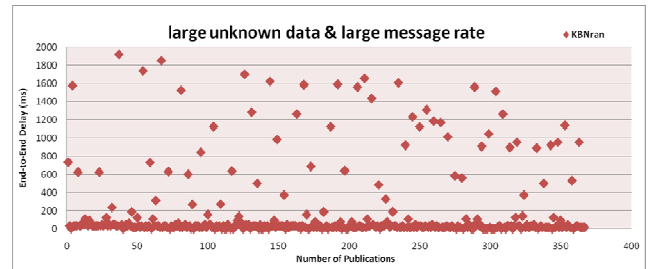
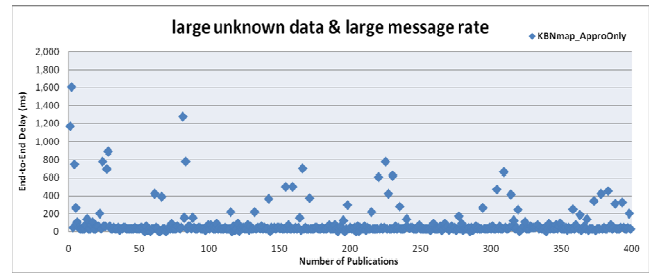


Figure 8: End-to-end time: High Unknown Data Rate with High Message Rate

*C. Case 2: Small Unknown Data Rate with Medium Mismatch Tolerance*

In this experiment we also evaluate the combination of two factors: the mismatch tolerance and the unknown data rate. While all combinations of these factors were tested, here we present a scenario where the unknown data rate was small (2 occurrences/minute) and the mismatch tolerance was medium. Again all other factors were set to default high-load settings as before. With these inputs the Bayesian Network produced a weighting of 1% for the *Every* strategy, 19% for the *ApproOnly* strategy, and 80% for the *ApproInd* strategy. As before, we compared the adaptive KBNMap implementation against the KBNRan implementation.

Again, as shown in Fig. 9 the KBNRan implementation executed the *Every* strategy early on in several routers, and continued to poor performance at random intervals during the experiment as the remaining brokers also executed the *Every* strategy. By comparison the KBNMap implementation selected the *ApproInd* strategy, which resulted in the strategy being executed more often, but with predictably low overhead for each execution.

As before we also measured the end-to-end delay of the subset of the publications that matched a subscription and so were routed across the entire network over 15 hops. From Fig. 10 it is again clear that KBNMAP performed better than KBNRan.

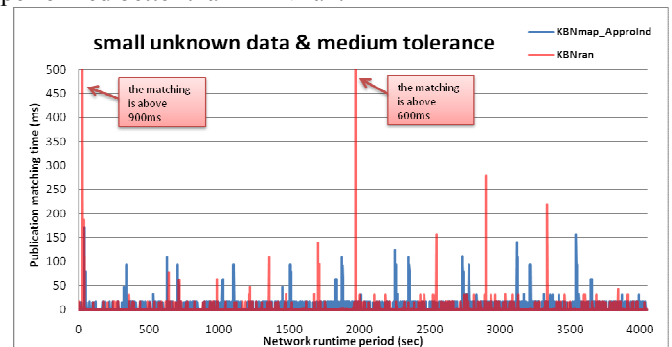


Figure 9: Pub Matching Time: Small Unknown Data Rate with Medium Mismatch Tolerance

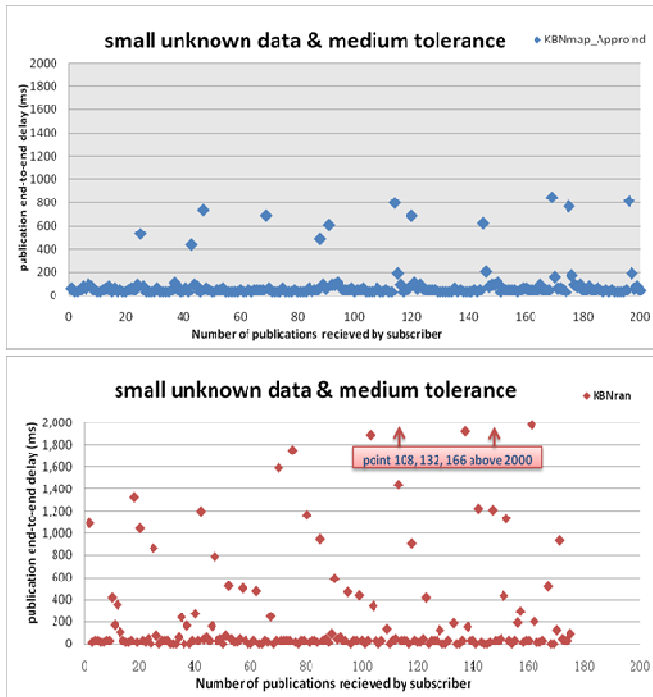


Figure 10: End-to-end Time: Small Unknown Data Rate with Medium Mismatch Tolerance

D. Case 3: Small Unknown Data Rate with High Memory Resources Available

As with the previous experiments we also evaluate the combination of two factors: the memory resources available and the unknown data rate. Again all combinations of these factors were tested, but here we present a scenario where the unknown data rate was small (2 occurrences/minute) and the available memory as set to high (120MB). Again all other factors were set to default high-load settings as before. With these inputs the Bayesian Network produced a weighting of 2% for the *Every* strategy, 23% for the *ApproOnly* strategy, and 75% for the *ApproInd* strategy. As before, we compared the adaptive KBNMap implementation against the KBNRan implementation.

Again, as shown in Fig. 11 the KBNRan implementation exhibited unpredictably high overhead early on in the experiment and then settled down to perform in a manner similar to KBNMap. Meanwhile KBNMap implementation again selected the *ApproInd* strategy, mainly due to the low number of occurrences of unknown data despite the availability of enough memory

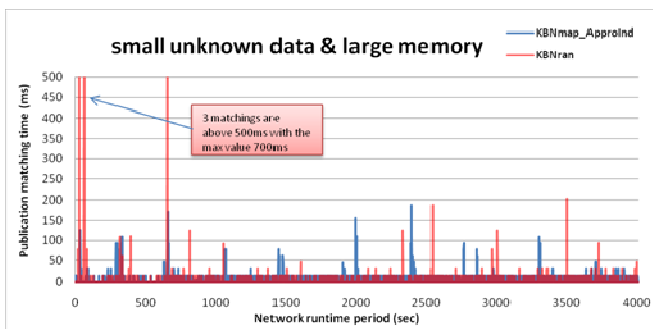


Figure 11: Pub Matching Time: Small Unknown Data Rate with Large Available Memory Resources

to execute the *Every* strategy. Again this resulted in the strategy being executed more often, but again with predictably low overhead for each execution.

As before we also measured the end-to-end delay of the subset of the publications that matched a subscription and so were routed across the entire network over 15 hops. From Fig. 12 it is again clear that KBNMAP performed better than KBNRan.

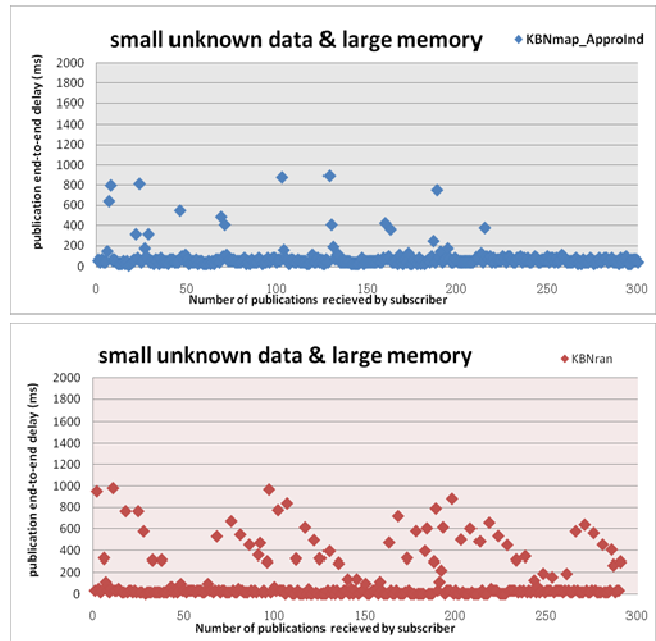


Figure 12: End-to-end Time: Small Unknown Data Rate with Large Available Memory Resources

X. CONCLUSIONS & FURTHER WORK

This paper presented a design, implementation and evaluation of a Knowledge-based Network extended to support semantic interoperability between multiple ontological knowledge-bases according to three alternative strategies. It is the first known such system to support multiple ontologies in this manner. We also identified the set of factors that influence strategy selection and when combined with a Bayesian Belief Network the KBNMap brokers adaptively and efficiently provides semantic interoperability services for resolving ontological heterogeneous data in a large scale network environment.

The experiments shown, and others omitted for space reasons, show that the adaptive and self-configuring mapping strategy management mechanisms are clearly superior to the randomly disordered mapping strategy mechanisms. Both the publication matching times and end-to-end publication latencies when handling unknown data are significantly reduced by implementing the adaptive strategy selection engine in the KBN broker. This also verifies our claim that the default weighting calculated for the combination of ontology, application, and environment level factors are adequate, however, we are confident that with the reconfigurable nature of the Bayesian Network, the factor nodes, states, and weighting support finer tuning for specific KBNMap deployments.

Further work can build upon these results, e.g., by considering other and also more general scenarios. It is also hoped that the semantic interoperability mechanisms and evaluation metrics can be directly embedded in other KBN implementations, thereby allowing direct comparison with the system presented here.

## REFERENCES

- [1] Meier, R., Cahill, V., "Taxonomy of Distributed Event-Based Programming Systems", *The Computer Journal*, 48(5), pp 602-626, 2005
- [2] Eugster, P. Th., Felber, P.A., Guerraoui, R., and Kermarrec, A.-M., "The many faces of publish/subscribe". *ACM Computing Surveys* 35(2) (2003) 114-131
- [3] Carzaniga, A., Rosenblum, D. S., and Wolf, A. L., "The Design and Evaluation of a Wide-Area Event Notification Service", *ACM Transactions on Computer Systems*, 19(3), Aug. 2001.
- [4] Segall, B. et al, "Content-Based Routing in Elvin4", In proc. AUUG2K, Canberra 2000.
- [5] Pietzuch, P., Bacon, J., "Peer-to-Peer Overlay Broker Networks in an Event-Based Middleware". DEBS'03 at ACM SIGMOD/PODS Conference. California, June 2003.
- [6] 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.
- [7] Strom et al., "Gryphon: An Information Flow Based Approach to Message Brokering", In proc. Intl. Symp. on Software Reliability Engineering 1998.
- [8] Berners-Lee, T., Hendler, J., Lassila, O., "The Semantic Web", *Scientific American*, May 2001.
- [9] W3C (2003) *Ontology Web Language*, <http://www.w3.org/2001/sw/>, Visited Mar 2009.
- [10] Wang, J., Jin, B., Li, J., "An ontology-based publish/subscribe system". In proc. *Middleware*, 2004.
- [11] Masuoka, R., Labrou, Yannis, Parsia, B., Sirin, E., "Ontology-Enabled Pervasive Computing Applications", *IEEE Intelligent Systems*, Sep-Oct 2004, pp 68-72.
- [12] Belecheanu, R., Jawaheer, G., Hoskins, A., McCann, J.A., Payne, T., "Semantic web meetings autonomic ubicomp" In proc. *Workshop on Semantic Web Technology for Mobile and Ubiquitous Applications*, Hiroshima, Japan, 2004.
- [13] Keeney, J., Roblek, D., Jones, D., Lewis, D., O'Sullivan, D., "Extending Siena to support more expressive and flexible subscriptions". In proc. DEBS 2008, Rome, Italy, July 2008
- [14] Keeney, J., Lewis, D., O'Sullivan, D., "Benchmarking Knowledge-based Context Delivery Systems". In proc. ICAS 06, Silicon Valley, USA, July 19-21, 2006.
- [15] 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.
- [16] Keeney, J., Lewis, D., O'Sullivan, D., "Ontological Semantics for Distributing Contextual Knowledge in Highly Distributed Autonomic Systems", *Journal of Network and System Management*, 15(1), March, 2007.
- [17] Keeney, J., Jones, D., Guo, S., Lewis, D., O'Sullivan, D., "Knowledge-based Networking", in *Handbook of Research on Advanced Distributed Event-Based Systems, Publish-Subscribe and Message Filtering Technologies*, [ed.s A. Hinze and A. Buchman], IGI Global, New York, 2009, in print.
- [18] Roblek, D. "Decentralized Discovery and Execution for Composite Semantic Web Services", M.Sc. Thesis, Computer Science, Trinity College Dublin, Ireland, 2006
- [19] Guo, S., Keeney, J., O'Sullivan, D., Lewis, D., "Coping with Diverse Semantic Models when Routing Ubiquitous Computing Information". In proc. MUCS2008 at NOMS 2008, Salvador, Bahia, Brazil, 11 April 2008.
- [20] Keeney, J., Jones, D., Roblek, D., Lewis, D., O'Sullivan, D., (2008), "Knowledge-based Semantic Clustering," In proc. ACM SAC 2008, Fortaleza, Brazil, 2008.
- [21] Keeney, J., Lewis, D., O'Sullivan, D., Roelens, A., Boran, A., Richardson, R., "Runtime Semantic Interoperability for Gathering Ontology-based Network Context". In proc. NOMS 2006, Vancouver, Canada. April 2006.
- [22] Tai, W., O'Sullivan, D., Keeney, J., (2008), "Distributed Fault Correlation Scheme using a Semantic Publish/Subscribe system," In proc. NOMS 2008, Salvador, Brazil, April 2008.
- [23] Petrovic, M., Burcea, I., Jacobsen, H.-A., "S-ToPSS: Semantic Toronto Publish/Subscribe System". In proc. VLDB03, Berlin, Germany, September, 2003.
- [24] Skovronski, J., Chiu, K., (2006), "Ontology Based Publish Subscribe Framework". In proc. *International Conference on Information Integration and Web-based Applications Services*, December 2006, Yogyakarta, Indonesia.
- [25] Cilia, M., Antollini, M., Bornhovd, C., And Buchmann, A., "Dealing with Heterogeneous Data in Pub/Sub systems: The Concept-Based Approach". In proc. DEBS04, Edinburgh, Scotland, 24 – 25 May 2004.
- [26] Lewis, D., Keeney, J., O'Sullivan, D., Guo, S., "Towards a Managed Extensible Control Plane for Knowledge-Based Networking". In proc. DSOM 2006, at Manweek 2006, Dublin, Ireland, October 2006.
- [27] McCann, R., Marcot, B.G., Ellis, R. "Introduction: Bayesian belief networks: application in natural resource management". *Can. J. For. Res.* 36(12), 2006.
- [28] "Bayesian Network", Available at: [http://en.wikipedia.org/wiki/Bayesian\\_network](http://en.wikipedia.org/wiki/Bayesian_network), Last Visit: January 2009
- [29] Marcot, B.G., Steventon, J.D., Sutherland, G.D., McCann, R.K., "Guidelines for Developing and Updating Bayesian Belief Networks for Ecological Modeling", *Can J. For. Res.* 36(12), 2006.
- [30] Cain, J "Planning Improvements in Natural Resources Management", Center for Ecology and Hydrology, Wallingford, Oxon., UK.
- [31] Guo, S., Keeney, J., O'Sullivan, D., Lewis, D., "Adaptive Semantic Interoperability Strategies for Knowledge Based Networking". In proc. SSWS '07 at OTM 2007, Vilamoura, Algarve, Portugal, November 2007
- [32] "Ontology Alignment Evaluation Initiative", Available at <http://oaei.ontologymatching.org/2007/>, Last visited: Mar 2009.