

# KNOWLEDGE-BASED NETWORKING

**John Keeney, Dominic Jones, Song Guo, David Lewis, Declan O’Sullivan**

*Knowledge & Data Engineering Group, Trinity College Dublin, Dublin, Ireland.*

Contact Author: John Keeney ([John.Keeney@cs.tcd.ie](mailto:John.Keeney@cs.tcd.ie))

Website: <http://kdeg.cs.tcd.ie/KBN>

## ABSTRACT

Knowledge-Based Networking, which is built on-top of Content-based Networking (CBN), involves the forwarding of events across a network of brokers based on subscription filters applied to some semantics of the data and associated metadata of the events contents. Knowledge-based Networks (KBN) therefore support the efficient filtered dissemination of semantically enriched knowledge over a large, loosely coupled network of distributed heterogeneous agents. This is achieved by incorporating ontological semantics into event messages, allowing subscribers to define semantic filters, and providing a subscription brokering and routing mechanism. The KBN used for this work provides ontological concepts as an additional message attribute type, onto which subsumption relationships, equivalence, type queries and arbitrary ontological relationships can be applied. It also provides a bag type to be used that supports bags equivalence, sub-bag and super-bag relationships to be used in subscription filters, composed with traditional CBN subscription operators or the ontological operators. When combined with the benefits of Content-based Networking, this allows subscribers to easily express meaningful subscription interests and receive results in a more expressive and flexible distributed event system than heretofore. Within this chapter the detailed analysis of ontological operators and their application to a publish/subscribe (pub/sub) domain will be fully explored and evaluated.

**KEYWORDS:** Distributed event-based Systems - Publish/Subscribe - Knowledge-based Network(s) - Content-based Network(s) – Semantic - Subscription Operators - Case-Studies/Scenarios - Semantic Event Filtering

## INTRODUCTION

Content-based networks (CBN), such as (Carzaniga et. al., 2001; Peitzuch et. al., 2002; Segal et. al., 2000), formed around the necessity to match a varying subscriber base to that of a network’s publication creators. However, open standards for CBNs have been slow to emerge due to the difficulty in reaching a general compromise between the expressiveness of event attribute types and subscription filters exasperated by the need to both match these efficiently at CBN nodes and to efficiently maintain routing tables. The need for efficient network utilisation requires that notifications are routed towards nodes and subscribers that are interested in a particular message, using a routing table composed of subscription filters, rather than flooding the network in the search for all interested possible parties. This is usually combined with a mechanism to exploit multicast efficiencies made possible by aggregating/covering subscriptions with ones that match the same or wider range of messages. For example, in the Siena CBN (Carzaniga et. al., 2001) subscription covering is achieved by restricting attribute types and subscription filters to simple number, string and boolean types using a set of transitive operators to filter across them (i.e. greater/less than, super/sub string etc.). The underlying routing structure on which messages pass allows for a message inserted on one side of the network to propagate across the network based on positive matches to the filters (subscriptions) until every client, and only those interested in the message, have been delivered the message.

CBN Subscriptions are specified as a set of filtering constraints constructed using a filter operator and some values used to compare against the contents of any incoming notification. The range of these operators determines the type of pub/sub network in which the message is being sent. Mühl et. al. (Mühl et. al., 2006) describe this content-based matching as a set of “Filters which are evaluated against the whole contents of notifications”. Within this work, notifications can be thought of as publications. Publications are only forwarded to a user when the contents of their subscriptions’ filtering constraints matches a subset of the message’s contents. This allows for a more flexible message format and, in comparison to topic-based networks, allows an even looser coupling between publishers and subscribers.

As an example, in the Siena CBN system (Carzaniga et. al., 2001) a notification is seen a set of typed attributes. Each attribute is comprised of a name, a type and a value. The existing version of Siena supports the following types: String, Long, Integer, Double and Boolean. A Siena subscription is a conjunction of filtering constraints, where constraints are comprised of the attribute name, an operator, and a value. A subscription matches a notification if the notification matches all the filtering constraints of the subscription’s filter. The notification is then delivered to all of the clients that submitted those subscriptions that match against that notification. Carzaniga (Carzaniga et. al., 2008) defined three basic types of Siena topology: hierarchical client/server, acyclic peer-to-peer, and general peer-to-peer. All topologies provide the same functionality, however they differ in non-functional features, like time complexity, scalability and fault tolerance.

The work within this chapter focuses on Knowledge-Based Networks (KBNs) (Lewis et. al., 2006; Keeney et. al., 2008a; Keeney et. al., 2008b) rather than CBNs. It is important to note that Knowledge-based networks are defined, within this chapter, as publish/subscribe networks in which the semantics of the message play an important part in the matching of publications to subscriptions. The particular flavour of KBN investigated in this work is an extension of the Java Siena CBN middleware (Carzaniga et. al., 2001). The specific Siena-based KBN implementation introduces two extensions to the existing type set and the set of filter operators already provided by the hierarchical Java version (Carzaniga et. al., 2008). These extensions add support for bags of values and the use of ontological types and operators. Specifically, the KBN implementation supports all Siena types in addition and union with ontological classes, properties and individuals (owl, 2004). In addition to this all of the standard Siena operators have added operators to exploit ontological type checking, subsumption, equivalence, disjointedness, and the use of arbitrary ontological object properties to capture relationships between individuals. The second extension examined in this chapter adds support for bags of values to be used in subscription filter constraints. Most CBN implementations only allow a single comparison value to be specified in each filter. Ince multiple constraints in a single subscription are usually combined as a conjunction, this greatly restricts the expressiveness of any single subscription filter. Rather than extending the subscription mechanism to support a disjunction of filtering constraints, which would greatly affect the ability to aggregate filters for efficient routing tables, our extension instead supports a disjunction of values within a single filtering constraint, with multiple constraints being combined as a conjunction as before.

Many of the additional capabilities of a KBN over a CBN could be achieved with complicated and unwieldy CBN subscriptions, which become unmanageable within both the network and with regard to the human creating the subscription. In addition, by extending the expressiveness of the subscription language we risk adversely affecting the potential for subscription aggregation in the broker network. However we have shown that more expressive subscriptions lead to more accurate and more concise subscriptions (Keeney et. al., 2007a), thereby improving performance and making a KBN more flexible and applicable in a wide range of use-cases.

## **RELATED WORK**

The use of ontological information in publish/subscribe systems is still on the edge of wide-scale deployment. Exploratory research and a detailed review of the State of the Art shows a gradual shift towards an understanding

of both the importance and power of semantics when combined with pub/sub networks. In this section we discuss seven influential systems showing similarities to the KBN implementation discussed here. For convenience, a comparison table is provided at the end of this discussion.

**S-ToPSS** (Petrovic et. al., 2003; Burcea et. al., 2003) offers a content-based network that is targeted at making an existing centralized syntactic matching algorithm semantically-aware whilst keeping the efficiency of current event matching techniques. S-ToPSS proposes a two-level distributed semantic pub/sub system. The top-level brokers have only high level descriptions of ontologies gained from the lower level routers. The lower level brokers maintain their own ontology for communication, heterogeneous ontological information for each application is distributed between multiple routers. These low level brokers advertise more general descriptions of the ontologies they hold to higher level brokers. Petrovic (Petrovic et. al., 2003) describes this additional semantic matching mechanism in an employer-employee scenario. The employer is looking for a candidate from a “certain university”, “with a PhD degree”, “with at least 4 years work experience.” This is matched using the semantically enhanced pub/sub system, S-ToPSS, in which an employee who has a PhD, from a particular school, with at least four years work experience is matched with the prospective employer. The match is only made because the semantic system is aware that “school” and “university” have the same semantic meaning, particularly in North American Educational System. S-ToPSS manipulates subscriptions as they arrive to add semantic synonyms and super-class concepts to the subscription (with support for more general mapping functions). In addition, when events enter the broker, new synthetic events are created depending on the semantics contained in the event. This way the semantic stage of subscription merging and publication matching is performed first, outside of the standard matching engine; unlike within a KBN. We argue that the KBN solution discussed in this paper is more general, flexible and expressive than S-ToPSS. More significantly, however, no report of an implementation or evaluation of this proposal has yet emerged.

An ontological pub/sub system called **Ontology-based Pub/Sub (OPS)** is presented in (Wang et. al., 2004), and shares our motivations to improve the expressiveness of events and subscriptions within the system. Wang’s work is achieved using RDF and DAML+OIL techniques to semantically describe events and subscriptions. Both of which are represented as RDF graphs and graph patterns respectively. The OPS system is shown to match events and subscriptions both semantically and syntactically. Central to the OPS matching algorithm is an efficient and scalable index structure based on the complete set of possible statement patterns (decomposed by RDF graph patterns) which are used as the basic unit of matching using AND-OR trees as matching trees, which subsequently avoids the backtracking of the RDF graphs. One criticism of the OPS system is that it does not include/provide the ability to perform generic content-based subscriptions. It is important to note at this point that the KBN documented within this chapter not only provides the semantic extensions documented, but couples these with the full set of Siena CBN operators, thereby allowing semantic and syntactic operators to coexist, increasing both the expressiveness and flexibility of the KBN.

The **Ontology-based Pub/Sub System (OBPS)** (Skovronski et. al., 2006) is another ontologically based publish/subscribe system. This system expresses event models by using the XML language: notifications are composed of an XML message where the tags are the names of the ontological classes or property within the ontology and the root node of the notification must contain the name of the ontology to which the notification is destined. To assist in the matching of events to the interested subscriptions, each publisher has its own publisher agent which is responsible for processing published events within the routing broker. Each agent inside the router maintains their own ontological model that stays in scope as long as the publisher continues publishing messages. The interested subscribers register a topic which defines the ontology in which they are interested, guaranteeing that all notifications for that topic are routed to all interested subscribers. The system uses the SPARQL query language as its subscription language, allowing the subscribers to easily parse the returned notification message based on their knowledge of which XML tags will be within the notification. Again like OPS, this system cannot

perform generic content-based subscriptions. All processing of messages and query analysis is done on the client-side. In addition, the substantial overhead introduced by SPARQL and the lack of mechanisms to aggregate subscriptions and construct network overlays means that the overhead and scalability of this system is uncertain.

**Continuous Querying Syndication System (CQS)** (Christian et. al., 2007) targets the formalisation of a syndication architecture that utilizes web ontologies and logic-based reasoning for a large volume of selective content dissemination on the web. This aims towards an expressive syndication system, where each of its subscriptions is comprised of a conjunctive ABox (ontology instance) query; while the publication is defined to be composed of a set of ABox assertions. Intuitively, CQS offers only one syndication broker which maintains a local knowledge base in which newly published information is integrated. To match newly published information with subscription requests in an efficient and practical manner, the events are matched to subscribers by employing a composition matching algorithm as follows: “information matches” refers to the instances that are stored in the local knowledge base bound to the variables of a continuous query representing a subscription. Hence the result returned to the interested subscriber is actually the query answer in the local knowledge base rather than the matched publications. Whilst “publication matches” refers to the collection of publications satisfying subscriptions, the broker delivers the minimal sets of publications to the interested subscribers.

**Elvin with Ontology (ElvinO)** is a proof-of-concept system implemented by the authors to investigate the feasibility of knowledge based networking (Keeney et. al., 2006b). The system uses closed-source Elvin (Segal et. al., 2000) CBN as an underlying subscription matching mechanism. In this system ontological synonyms were added to publications at the edge of the network in a generic mapping gateway, allowing subscribers to register subscriptions according to their own ontology, and have the matching publications delivered to them, where the publication may have contained information defined in a different (mapped) ontology. Alongside the semantic subscriptions, this system maintains the very expressive Elvin subscription language thereby supporting semantic and content based subscriptions. Additionally, the Elvin subscription language is more expressive than that provided by Siena, in particular supporting logical combinatorial operators, thereby additionally supporting combinations and disjunctions of constraints. The closed-source nature of the Elvin implementation is also prohibitive in terms of enhancing the content based routing algorithms to include more complex datatypes, such as OWL classes instances or properties inside the broker rather than at the edge of the network.

**The Semantic Pub/Sub with Super-Peers (SPS-SP)** (Chirita et. al., 2004) system is an RDF-based system for managing arbitrary digital resources. Subscription queries are expressed by using a typed first-order RDF query language called “L”. A subscription is a conjunction of RDF triples, where an RDF triple is comprised of a subject (s), predicate (p), and an object (o). The predicate can be the operator of “>” for integer type or “ $\supseteq$ ” meaning “contains” for string type. The publication is a pair (T, I), where T is a set of ground atomic formulas of L of the form of an RDF Triple (s, p, o) with the same constant s, and I being a client identifier. A publication PUB matches a subscription SUB if the  $\text{ans}(\text{SUB}, T) \neq \emptyset$ ; where the  $\text{ans}(\text{SUB}, T)$  denotes the answer set of SUB when it is evaluated over T. Publications and subscriptions are matched at super-peers and appropriate subscribers are notified. The advertisement of messages is also used to help super-peers route information efficiently. The SPS-SP system is comprised of peers with the role of publisher or subscriber and a super-peer that offers a more powerful capability than a typical peer. Super-peers are organized within the HyperCup (Schlosser et. al., 2002) semantic topology which takes care of processing publications, advertisement and subscriptions using P2P semantic queries. The HyperCup algorithm is capable of organizing super-peers into a binary hyper cube that is a type of Cayley graphs. The advantage of HyperCup is that it limits the path between any two super-peers to  $\log_2 N$ , which enables efficient and non-redundant information broadcasting. The SPS-SP system utilizes a selective advertisement forwarding algorithm to selectively broadcast advertisement messages to super-peers. Therefore, the peers that submit subscriptions can exploit the advertised information to calculate

which peers contains information that should match their subscriptions. Again, like several of the systems discussed, this system does not include capability to perform generic content-based subscriptions.

## Other Relevant Work

Semantic web technologies aid a user in the continued search for relevant information (Grossnickle et. al., 2005) shows how the introduction of semantically rich meta-data to a rapidly changing source of RSS news feeds increases the ease in which searched information is delivered to a user. This is supported by Borsje et al. in (Borsje et. al., 2008) using a localised SPARQL query. This query could be represented in pub/sub terms, as a user's subscription. Knowledge-distribution systems deliver semantically marked-up data to be delivered to interested parties in both a timely and efficient manner. This is achieved by embedding ontological mark-up of the feeds. However, with the substantial overhead of using SPARQL as the query mechanism and the lack of subscription aggregate techniques, the overhead and scalability of this system is unclear.

Baldoni (Baldoni et. al., 2007) supports the rationale behind the introduction of knowledge representation through the use of tagging each publication within the network with a topic. This can be likened to assigning a subject line to an email, except in this case the message is instead assigned a topic tag. The subscription table is constructed using a list of couples  $\langle t, i \rangle$  where  $t$  is the topic a node is subscribed to and  $i$  is the corresponding topic overlay identifier. Upon retrieving a new subscription for a topic, the subscription management component adds an entry for the topic to the subscription table and then passes the task of connecting the corresponding overlay networks. In the tagging of messages it is easy to see that this increased level of descriptiveness is beneficial in the task of matching messages.

Li (Li et. al., 2004) presents a semantic-based system based on a centralised pub/sub bus implementation. Its application is limited to enterprise scale deployment, not fully offering true CBN capabilities. Li offers approaches to the enhancement of subscriptions and events within the proposed semantically enhanced system. Li does not, however, address the issues involved in introducing ontologically based reasoning within a distributed event dissemination network and does not offer an implementation or evaluation of the proposed system.

The pub-sub middleware presented by Cilia (Cilia et. al., 2003) allows semantics to be used in the publications and subscriptions. However, these semantics are used on the edge of the network in a manner similar to that presented in number of systems above.

Semantic techniques are also being used by several systems for the retrieval of distributed ontologically encoded knowledge in P2P DHT systems (Tempich et. al., 2004; Cai et. al., 2004; Loser et. al., 2003). These systems, such as the one presented in (Borsje et. al., 2008), focus on query-response communication as opposed to the pub/sub model presented within this chapter.

As shown by Meier (Meier et. al., 2005) and Eugster (Eugster et. al., 2003) there are various type- and topic-based/subject-based distributed event-based systems whose popularity has both increased and subsequently decreased over time. Several of these systems use hierarchical addressing in the organisation of topics using a containment relationship in a manner that is loosely similar to the taxonomical class hierarchy defined in an ontology. The mechanisms addressed by these systems allows the use of the notion of sub- and super-class subscriptions with regard to the topic hierarchies, which in this respect allow a simple comparison to KBN subscription operators. However the subscriptions using topic-based hierarchical addressing only operate in an "up/down" subscription creation algorithm. With the use of equivalence and disjoint relationships between classes, a KBN allows a multidimensional type tree to be created in which sub-classes not only point to classes

further down the class hierarchy, but can point to other classes in other parts of the ontology, a unique addition to the traditional topic-based subscription.

XML-based systems (Diao et. al., 2003; Gupta et. al., 2003; Chan et. al., 2006) publish events as XML documents where XPath (Clark et. al., 1999) queries allow expressive and flexible subscriptions, and arbitrarily complex queries are applied to a DOM tree derived from the published XML document. The message is then forwarded only when a match is found. XML-based systems provide an increased level of expressiveness in comparison to topic-based systems. However, the message architecture is exclusively based on tree patterns and thus is less expressive and flexible than KBN messages.

System	Event Model	Subscription Language		Overlay Type	Routing Scheme		Event Matching
	<b>Data Expression:</b> <i>The format of knowledge within the network.</i>	<b>Subscription Expression:</b> <i>How user subscriptions are formed.</i>	<b>Subscription Operators:</b> <i>The operators used in the user subscriptions.</i>	<i>The overlay network design and deployment</i>	<b>Sub Forwarding:</b> <i>The method in which a broker forwards subscriptions to other nodes in the network</i>	<b>Event Forwarding:</b> <i>How the pubs are forwarded to interested subs</i>	<b>Pub to Sub Matching:</b> <i>How incoming pubs are matched to stored subscriptions.</i>
KBN	<b>Basic Format:</b> [attribute_name, value] <b>Non-ontological_value:</b> Integer, Long, Double, Boolean, String <b>Ontological_value:</b> OWL Classes, Properties, and Instances.	<b>Basic Format:</b> [attribute, operator, value] <b>Note:</b> Similar to Siena	<b>Non-ontological:</b> >, <, =, ≤, ≥, etc. <b>OWL operators:</b> equivalent, subsumes, subsumed by, instance_of, object property <b>Bag operators:</b> (e.g., subbag, superbag)	<b>Hierarchical structure</b> <b>Semantic-based cluster</b>	<b>Sub Aggregation method:</b> aggregated by subscription covering. <b>Forwarding method:</b> The most general subscriptions are hierarchically sent to master servers.	<b>Reverse path forwarding:</b> follows the reverse path of subs.	<b>Matching tree:</b> Event iterates through sub tree using breadth first search.
S-ToPSS	<b>Basic Format:</b> [attribute_name, value] <b>Non-ontological_value:</b> Integer, Long, Double, Boolean, String <b>Ontological_value:</b> uses three extra matching algorithms to make a non-ontological event semantically.	<b>Basic Format:</b> [attribute, operator, value] <b>Note:</b> Similar to ToPSS	<b>Crisp:</b> >, <, =, ≤, ≥, etc. <b>Approximate:</b> (e.g., ≈ modelling approximate equality) <b>Probability:</b> (e.g., modelling a random event)	<b>Centralized:</b> The system is deployed as an information dissemination service	<b>None:</b> it is a centralized, therefore no routing	<b>None:</b> it is a centralized, therefore no publication forwarding	<b>Matching tree:</b> Event iterates through sub tree using breadth first search. <b>For semantic matching:</b> Existing matching assisted with: -Synonyms matching -Concept matching -Hierarchy matching
OPS	<b>Basic Format:</b> [attribute_name, value] <b>Non-ontological_value:</b> Integer, Long, Double, Boolean, String <b>Ontological:</b> RDF graph (concepts within Events form an ontology)	<b>Basic Format:</b> RDF graph pattern (subject, object, meta-statement, [filter_func(object)])	<b>Note:</b> When the object is a variable and its type is literal, it includes <i>filter_func(object) to refine value</i> <b>filter:</b> >, <, =, ≤, ≥, etc	<b>Hierarchical structure</b> <b>Peer-to-peer</b>	<b>Aggregation method:</b> Aggregation by graph merging. <b>Forwarding method:</b> the most general subscriptions are sent to parent servers.	<b>Reverse path forwarding:</b> Follows the reverse path of subs.	<b>AND-OR matching tree:</b> Node and arc in sub graph must be mapped to node and arc in the event graph.
OBPS	<b>Basic Format:</b> [attribute_name, value] <b>Non-ontological_value:</b> Integer, Long, Double, Boolean, String <b>Ontological_value:</b> XML-based (Tags represent the name of classes & properties of an ontology model)	<b>Basic Format:</b> SPARQL query.	<b>No operators:</b> subscription is represented by SPARQL query.	<b>Centralized:</b> a single router	<b>None:</b> it is a centralized, therefore no routing	<b>None:</b> it is a centralized, therefore no publication forwarding	<b>Ontological topic based matching using a SPARQL engine.</b>
CQS	<b>Basic Format:</b> $(a, t, p)$ where $a$ is a set of DL ABox assertions, $t$ is time unit and $p$ is the client identifier	<b>Basic Format:</b> $(Q, t)$ where $Q$ is a Continuous conjunctive ABox query with respect to a DL KB and $t$ is time unit	<b>No operators:</b> Subscription is represented by ABox query.	<b>Centralized:</b> a single syndication broker	<b>None:</b> it is a centralized, therefore no routing	<b>None:</b> it is a centralized, therefore no publication forwarding	<b>Composition matching algorithm:</b> -Information matches. - Publication matches. -Composite matches.
Elvin(O)	<b>Basic Format:</b> [attribute_name, value] <b>Non-ontological_value:</b> Integer, Long, Double, Boolean, String.	<b>Basic Format:</b> [Attribute, operator, value]	<b>Non-ontological:</b> >, <, =, etc. <b>Logic:</b>  , &&, !, etc.	<b>Clustering:</b> local area server replication for fault tolerance <b>Federation:</b> static wide area hierarchical subscription space partitioning	<b>Forwarding method:</b> Subscription kept locally, possibly broadcast within clusters (limited information available)	<b>Hybrid:</b> Between federations publications are routed according to subscription space partitioning, possibly broadcast within clusters (limited information available)	<b>Non-ontological:</b> unknown, internal to Elvin <b>String based label Ontological matching:</b> OWL labels and synonyms are merged into/from string labels at the edge of network.

SPS-P2P	<p><b>Basic Format:</b> <math>(T, I)</math> where <math>T</math> is a set of <math>t(s, p, o)</math> and <math>I</math> is a client identifier</p> <p><b>Ontological_value:</b> RDF Event Model</p>	<p><b>Basic Format:</b> <math>(s, p, o)</math> where <math>s</math> is a variable, <math>p</math> is constant, <math>o</math> is distinct variable</p> <p><b>Note:</b> Datalog-inspired RDF Query</p>	<p><b>Binary:</b> <math>&gt;, &lt;, =, \leq, \geq</math>, etc.</p> <p><b>Subsumes:</b> <math>\supseteq</math></p>	<p><b>Peer-to-Peer</b></p> <p><b>Note:</b> Super-peer arranged in HyperCup topology.</p>	<p><b>Aggregation method:</b> Aggregation by subscription covering (similar to SIENA).</p> <p><b>Forwarding method:</b> the most general subscriptions are broadcasted to all neighbours</p>	<p><b>Reverse path forwarding:</b> Follows the reverse path of subs (similar to SIENA).</p>	<p><b>Matching tree:</b> The event iterates through the subscription tree by using breadth first search algorithm (similar to SIENA).</p>
---------	---	---	---	--	--	---	---

## THE SEMANTIC EXTENSION

Knowledge-based Networking, an extension to Content-based Networking, involves the routing of events across a network, based not just on the values of the event contents but also on some semantics of the data and associated meta data contained in the event. We have developed a model for the filtered dissemination of semantically enriched knowledge over a large loosely coupled CBN of distributed heterogeneous agents. We call such a semantic-based CBN a Knowledge-Based Network (KBN). In (Lynch et. al., 2006; Lewis et. al., 2006; Keeney et. al., 2007a; Keeney et. al., 2008b) a KBN implementation is presented that extends Siena by providing three additional ontological base types: properties, concepts/classes and individuals/instances, as described in ontologies originating from the semantic web community. It also supports subsumptive subscription operators, i.e. sub-class/property (MORESPEC) i.e. *more specific*, super-class/property (LESSSPEC) i.e. *less specific*, and semantic equivalence (EQUIV). For example, as seen in the Wine ontology (w3c, 2003), the ontological type “wine” is less specific than (subsumes) the type “white wine”, as “white wine” is more specific than “wine” since “wine” is a superclass of “white wine”. Producers of knowledge express the semantics of their available information based on the ontological representation of that information. Consumers express subscriptions upon that information as simple semantic queries. If an event consumer was interested in receiving events about some ontological entity E, classes equivalent to E, or entities more specific than E, this can be easily achieved by creating a filtering constraint such that the entity described in a field x of the message is subsumed by E, i.e., (x MORESPEC E). E.g. A subscriber can subscribe to all KBN messages that contain an attribute whose value is a concept more/less specific than the named concept in the subscription. This approach provides loose semantic coupling between applications, which is vital as a new wave of applications increasingly rely on using the application information, context and services offered by existing heterogeneous distributed applications. To achieve this, each KBN router holds a copy of a shared OWL ontology (owl, 2004), within which each ontological class, property and individual used is described and reasoned upon.

The new ontological types, namely classes, individuals, and properties, are first class KBN types and can be used in any KBN subscription or notification, along-side the standard CBN (Siena) types and operators. This allows messages to be matched to subscriptions based on extensible type information, which can effectively represent meta-data for the message without having to maintain an ever-growing set of universal attribute names. Instead, a simple set of shared attribute names can be used for a concept type, which uses values from a taxonomy that is maintained, distributed and reasoned over at run-time using existing standardised ontology techniques. In addition to the equivalent class, equivalent property, equivalent individual (EQUIV) operator; subclass, subproperty (MORESPEC) operator; and superclass and superproperty (LESSSPEC) operators we also discuss the following operators: ISA, IS\_NOT\_A, ONTPROP, and NOT\_EQUIV.

The ISA operator is used to match an ontological individual/instance against its ontological types/classes. If an individual I is defined as being of an instance of a certain type C then the ISA operator will match the individual I to class C, all classes equivalent to C, and all superclasses of C. For example if the person “John” is represented in an ontology as an individual of the class “person”, and the class “person” is defined to be equivalent to the class “human”, and the class “human” is a subclass of the class “mammal”, then the individual “John” is related by the ISA relationship to the classes “person”, “human”, and “mammal”. So the subscription filter ( x ISA “human”), where x is an ontological individual, would match a notification that contains the named attribute ( x : “John”). This kind of a subscription filter was not previously possible since the EQUIV, MORESPEC, and LESSSPEC operators could only compare classes with classes, properties with properties, and individuals with individuals.

The IS\_NOT\_A operator is again used to compare an ontological individual/instance against its ontological types. If an individual I is defined as being of an instance of a certain type C then the IS\_NOT\_A operator will

match the individual I to all classes except class C, all classes equivalent to C, and all superclasses of C. Based on the example above the individual “John” is a “human” and is a “mammal”, but is not a “cow” where “cow” is a subclass of “mammal”. So the subscription filter (  $x$  IS\_NOT\_A “cow”) would match a notification that contains the named attribute (  $x$  : “John”).

The ONTPROP operator is used to match ontological individuals against each other using any ontological object property. Ontological object properties define named relationships between individuals of two classes. For example the object property “eats” might be defined between individuals of type “animal” and individuals of type “food”, so the individual called “Colleen” of type “cow” (“cow” is a subclass of type “animal”) would be related by this “eats” relationship to an individual called “grass” of type “food”. So the filter (  $y$  ONTPROP<sup>“eats”</sup> “grass”), where  $y$  is an ontological individual, would match a notification that contains the named attribute (  $y$  : “Colleen”).

Siena does not support a generic NOT (!) operator modifier, and so the NOT\_EQUIV operator was added for completeness. This operator is the opposite of the ontological EQUIV operator discussed above. It is used to compare classes with classes, properties with properties, and individuals with individuals.

## THE BAG EXTENSION

According to Weisstein (Weisstein, 2002), a bag (also called multiset) is a set-like object in which order is ignored, but multiplicity is explicitly significant. Therefore, bags {1, 2, 3} and {2, 1, 3} are equivalent, but {1, 1, 2, 3} and {1, 2, 3} differ. A bag differs from a set in that each member has a multiplicity indicating how many times it is a member.

As presented by Roblek (Roblek, 2006) a bag value can contain any valid Siena/KBN values, including other bag values. A bag is not allowed to contain itself, either directly or indirectly via other bags. Elements of a bag do not need to be of a uniform type. In the extension presented here bags are first order members of the Siena/KBN type set. They can appear in notifications as well as in subscription filters like any other Siena/KBN type. Siena advertisements, which are part of the theoretical Siena model, are not supported in the hierarchical version of Siena but, should they be, bag type should work seamlessly with them.

Therefore, some examples bags are: {3, 345, 27, 35, 3476, 0, 27, 27}, {"Ljubljana", 2, "Ljubljana", 3.14159}, {"Ljubljana", "Vienna", "Amsterdam", "Dublin"}. Since a set is a bag where all elements have a cardinality of one, this extension also implicitly supports sets. The bag extension adds simple binary bag operators and composite binary bags operators.

### Simple Bag Subscription Operator

The simple operators support the three well-known binary bag relations: equal, subbag, and superbag. Two bags A and B are equal ( $A=B$ ), if the number of occurrences of each element in A or B is the same in each bag. For example the bags {'b', 'o', 'o', 'k'} and {'b', 'o', 'k', 'o'} are equal bags but {'b', 'o', 'o', 'k'} and {'b', 'o', 'k'} are not equal bags.

For the subbag relationship, we can define A to be a subbag of B,  $A \subseteq B$ , if the number of occurrences of each element  $\chi$  in A is less than or equal to the number of occurrences of  $\chi$  in B. For example {'b', 'o', 'k'}  $\subseteq$  {'b', 'o', 'o', 'k'} but {'b', 'o', 'w'} is not a subbag of {'b', 'o', 'o', 'k'}. It follows from the definition of subbag that bags A and B are equal bags, if and only if  $A \subseteq B$  and  $B \subseteq A$ . Another way to describe the subbag relationship is to say that if  $A \subseteq B$  then bag B includes or contains all of the elements in A. The superbag relationship is the inverse of the subbag relationship. If B is a subbag of A,  $B \subseteq A$ , then A is a superbag of B,  $A \supseteq B$ . Also note, a bag can contain other bags. For example {'g', 'o', 'o', 'd'}, {'b', 'o', 'o', 'k'}  $\supseteq$  {'b', 'k'}

{‘o’, ‘o’}}. All three simple bag relations, namely equal, subbag, and superbag, are transitive and reflexive. The transitivity and reflexivity of simple bag relations follows from the transitivity and reflexivity of numerical *equal* and *less than or equal* relations that were used in the definition of the simple bag relations.

The main advantage of the bag type and bag operators lies in the ability to define much more expressive and flexible subscriptions. For example, without the use of bags, multiple filter constraints in a single subscription filter are joined by conjunction (using the boolean AND operator). A disjunction of constraints (using the boolean OR operator) could only be specified using multiple distinct subscriptions. Using bags and the subbag operator the subscriber can create a filter where matching values should be drawn from a bag of possible values for example. A particularly useful example of this is in Keyword matching.

## Composite Bag Subscription Operator

While the simple bag operators defined above prove useful for a number of case studies, they are restricted to comparing the sizes and structures of bags rather than performing more useful comparisons over the *elements contained* in the bags. Using the simple bag operators bags are only compared based on the presence of *matching* elements in each bag. For example, a subscriber may be interested in a bag of strings that is larger than that specified in her subscription (superbag), but the strings used in her subscription bag should be substrings of those in matching publications rather than equal strings. The simple bag operators can also be combined with other Siena KBN operators to produce composite bag operators. The composite bag relation is also a binary relation over bags, but is composed of (i) a simple binary bag relation over the bags and (ii) a sub-relation over the bags’ elements. In this way two bags can be compared not just on the presence of *matching* elements but rather by using more expressive operators to compare the elements that make up each bag.

Suppose  $\Phi$  is a simple binary relation over bags ( $\Phi$  is a simple bag operator), and  $\lambda$  is an arbitrary binary relation ( $\lambda$  is *any* non-bag subscription operator). Bag P is  $\Phi$ -related to bag Q when sub-relation  $\lambda$  is applied between the elements of P and Q, written as  $P \Phi_\lambda Q$ , if and only if there exist some sequences X and Y, (X is some ordered list of elements from P, and Y is some ordered list of the elements from Q), so that all of the following statements are true:

1. P is  $\Phi$ -related to  $\tau(X)$ , where  $\tau(X)$  denotes the bag of all elements in sequence X.  
*bag P is  $\Phi$ -related to X when X is expressed as a bag*
2.  $\tau(Y)$  is  $\Phi$ -related to Q, where  $\tau(Y)$  denotes the bag of all elements in sequence Y.  
*when Y is expressed as a bag, that bag is  $\Phi$ -related to Q*
3.  $|X| = |Y|$   
*sequences X and Y have the same number of elements*
4.  $\forall i \in \mathbb{N}$  (natural number),  $i < |X|$ ,  $X_i$  is  $\lambda$ -related to  $Y_i$ .  
*for (int i = 0; i < |X|; i++)*  
*X.elementAt(i) is  $\lambda$ -related to Y.elementAt(i)*

We call relation  $\Phi$  the primary relation of the composite bag relation, and relation  $\lambda$  the sub-relation of the composite bag relation. So for bags P and Q, simple bag relation  $\Phi$ , and any relation  $\lambda$ ,  $P \Phi_\lambda Q$  means that P is  $\Phi$ -related to bag Q when sub-relation  $\lambda$  is applied. If the composite  $\Phi_\lambda$  relation is being used as a subscription operator we call  $\Phi$  the primary bag operator of the  $\lambda$  the suboperator.

The bag of integers {1, 1, 2, 3, 4} is a superbag of {2, 4, 3} using the default “equals” (=) sub-relation, i.e. {1, 1, 2, 3, 4}  $\supseteq$  {2, 4, 3} (for every element in the second bag, there exists an element in the first bag that is equal to the element, with no reused elements in either bag).

The bag of integers  $\{1, 2, 3\}$  is an equal-bag of  $\{2, 3, 4\}$  using the “less than” ( $<$ ) sub-relation (*for every element in the second bag, there exists an element in the first bag that is less than the element, with no unused or reused elements in either bag*), so  $\{1, 2, 3\} =_{<} \{2, 3, 4\}$ .

The bag of Strings  $\{\text{“ood”}, \text{“boo”}\}$  is a sub-bag of  $\{\text{“a”}, \text{“good”}, \text{“book”}\}$  using the “substring” (substr) sub-relation (*for every element in the first bag, there exists an element in the second bag such that the element in the first bag is a substring of the element in the second bag, with no reused elements in either bag*), so  $\{\text{“ood”}, \text{“boo”}\} \subseteq_{\text{substr}} \{\text{“a”}, \text{“good”}, \text{“book”}\}$ .

Note, the simple bag operators defined in the previous sections can be defined as composite bag operators using the default “equals” ( $=$ ) sub-relation. More generally, any simple bag operator  $\Phi$  is equivalent to the  $\Phi_{=}$  composite bag relationship.

Where  $\Phi$  be a simple binary relation over bags, and  $\lambda$  a binary relation over bag elements. (i.e.  $\Phi_{\lambda}$ ,  $\Phi$  is the primary bag operator,  $\lambda$  is the sub operator). If  $\lambda$  is transitive, then  $\Phi_{\lambda}$  is also transitive. If  $\lambda$  is reflexive, then  $\Phi_{\lambda}$  is also reflexive. Another interesting observation is that if  $A \subseteq_{>} B$  then  $B \supseteq_{<} A$ . More generally, if  $\Phi$  is a simple binary relation over bags,  $\lambda$  is a binary relation over bag elements,  $\Phi^{-1}$  is the inverse relation of  $\Phi$ , and  $\lambda^{-1}$  is the inverse relation of  $\lambda$ , then  $P\Phi_{\lambda}Q$  exactly when  $Q\Phi_{\lambda^{-1}}^{-1}P$ . This observation only holds where inverse subscription operators exist, for example, some operators have no inverse (e.g. substring, some ontological properties), and as mentioned, Siena and the presented KBN implementation does not support a generic boolean NOT (!) operator modifier.

Of further note, a composite bag operator may have another composite bag operator as its sub operator. For example given

$$V = \{\{\}, \{0, 0\}, \{1, 2, 3, 4\}\}$$

$$W = \{\{8\}, \{0\}, \{1, 1, 1\}, \{2, 3, 4, 5, 6\}\}$$

then

$$V \subseteq_{(\subseteq_{<})} W$$

since

$$\{\} \subseteq_{<} \{8\}, \{0, 0\} \subseteq_{<} \{1, 1, 1\},$$

$$\{1, 2, 3, 4\} \subseteq_{<} \{2, 3, 4, 5, 6\}, \text{ and bag } \{0\} \text{ in } W \text{ is unused.}$$

Compared to simple bag relations, composite bag relations make looser comparisons of bags possible. They allow for “inexact” matches that would not be possible should we use only simple bag relations. Several content-based publish/subscribe systems support disjunction type subscriptions and the use of sets in the formation and matching of subscriptions. However, the Bag extension presented within this chapter (and combined with ontological subscription formation) allows Siena to now support such queries. However we argue that the composite bag operators supported by our KBN surpasses this with the ability to define even more expressive and flexible subscriptions, especially when the composite bag operators (unique to this work) are used, particularly when used together with the ontological types and operators.

## Performance of Subscription Tree Merge and Lookup using Simple Bag Operators

Suppose there exists some bags  $B$  and  $C$ . Without loss of generality we can assume that  $|B| \leq |C|$ . The simple bag operator comparison has the best time complexity  $\mathbf{O}(|B|)$  and the worst time complexity  $\mathbf{O}(|C|^2)$ . If bags are pre-ordered it is possible to use a better algorithm that has the worst time complexity of  $\mathbf{O}(|C|)$ , assuming the bag elements can be totally ordered. Since sorting a list of  $n$  element has an average complexity of  $\mathbf{O}(n \log n)$ , with both bags requiring pre-sorting, we did not opt to explicitly pre-sort each bag before attempting bag comparison.

This decision should have a relatively small performance affect unless the bags being compared are particularly large. In the current KBN implementation bags are either pre-ordered or not depending on the order in which values are encoded into subscriptions or publications by the subscribing/publishing application.

### Performance of Subscription Tree Merge and Lookup using Composite Bag Operators

With the brute force algorithm used to compare bags with the composite bag operator, in the most optimistic case the algorithm finds matching elements immediately. In this case the time complexity of the algorithm is  $O(|B|)$  for some bags B and C where  $|B| \leq |C|$ .

In the most pessimistic case the algorithm must fully match all possible arrangements of the elements of the smaller bag with the elements of the bigger set. In this case the time complexity is:

$$O\left(\frac{|C|!}{(|C|-|B|)!}\right) < O(|C|^{|B|})$$

We can see from the formulas above that the time complexity depends on the size of the smaller bag because the size of the smaller bag appears in the exponent of the time complexity. It is evident that the algorithm for composite bag operators implemented in the scope of this research is truly useful only for bag comparisons where at least one of the bags is small.

It is possible to develop much more effective comparison algorithms for certain specialized composite bag operators. For example, as discussed above it is possible to exploit the ordering of the elements to improve the algorithm. For composite bag operators over bags containing numbers, where the sub-operator is one of the  $<$ ,  $\leq$ ,  $>$ ,  $\geq$  it is possible to develop a very effective comparison algorithm if bags are pre-ordered. This is because the set of all integers is a totally ordered set with regard to  $\leq$ , or  $\geq$ . Such an algorithm would have a very low time complexity of  $O(\max(|B|,|C|))$ . Unfortunately not all sets are totally ordered, or can be totally ordered. For example, ontological instances cannot be naturally ordered, ontological concepts form only a partially ordered set with regard to subsumption. An effective algorithm for composite bag operators over partially ordered sets could be a subject of the future research. Alternatively, there might exist some constraints that narrow the set of all possible bags that are to be published in notifications or subscription filters, to some subset, for which good performance of the composite bag operator could be guaranteed.

### IMPLEMENTING A KBN

The two extensions discussed in this paper have been fully implemented and tested, and a deployable Java KBN implementation is available. Though our initial measurements described in (Keeney et. al., 2006b) used the Elvin CBN, this was a centralised system and our scalability goals required us to consider a decentralised CBN scheme. For the design and implementation presented here we opted to build upon the Siena CBN (Carzaniga et. al., 2001) due to source code availability and an abundance of associated technical reports and papers, and in addition, its focus on expressiveness in a wide-area distributed environment. The approach taken was to extend the CBN to use ontological reasoning using Jena (Carroll et. al., 2004) for ad hoc matching and filtering inside the network. The open-source nature of Siena also allows us to migrate the semantic subscription and message matching from the endpoint network nodes to within the network itself.

Previous works by the authors have shown that semantic types and operators can be supported by incorporating an ontological knowledge base and an ontological reasoner into each Siena KBN router/broker (Lynch et. al., 2006; Lewis et. al., 2006; Keeney et. al., 2007a; Keeney et. al., 2008b). Each KBN router/broker holds a copy of a shared OWL ontology, within which each ontological class, property and individual used is described and reasoned upon. To achieve this there have been significant additions to the codebase to support the

new ontological types and operators. However, it must be noted that the operation of the described KBN as a non-semantic Siena CBN has not been compromised in any way. The semantic extensions (and bag extensions) are provided to supplement the Siena CBN system.

Previous works by the authors have also shown that the loading of new ontologies into a reasoner embedded in a KBN node is computationally expensive due to load-time inference (Lewis et. al., 2006). Therefore the frequency of changes to the ontological base of a given KBN must be minimised since changes will need to be distributed to each of the nodes in the network. Secondly, ontological reasoning is memory intensive and memory usage is proportional to the number of concepts and relationships loaded into the reasoner so reasoning latency can be controlled by limiting this number in any given KBN node. However, once loaded and reasoned over, the querying of such an ontological base is relatively efficient with performance relative to size of the ontological base (Lewis et. al., 2006).

A large number of ontology reasoners are available, including: KAON2 (Motik et. al., 2006), Pellet (Parsia et. al., 2004), Racer (Haarslav et. al., 2001), FaCT (Tsarkov et. al., 2005), and F-OWL (Zou et. al., 2004). Any choice of a reasoner must be based on examination of performance evaluations in the literature, such as (Pan, 2005; Guo et. al., 2004; pellet, 2003; Motik et. al., 2006), as well as with separate benchmarking. These evaluations must also be compared to the performance characteristics of domain specific reasoners, or existing reasoners cut-down to give reduced but sufficient results in return for enhanced performance. This trade-off around reasoning performance versus expressiveness and accuracy of the model, after the inference cycle, is of particular importance where such reasoning may be required for efficient and correct routing in the network. The differing performance characteristics of different reasoners under different conditions, such as, the impact of the ratio of concepts to relationships of subsumption relationships to user defined predicates, must also be evaluated. 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) (Pan, 2005; GuoY et. al., 2004; GuoY et. al., 2005; pellet, 2003; Motik et. al., 2006). Of particular importance is the amount of reasoning that can be performed at ontology load time versus when the first or subsequent queries are submitted to the ontology. While the load time overhead of incorporating an ontology and a reasoner at every broker is substantial, we have shown that the runtime overhead of querying the ontology is minimal when performing subscription merging or publication/subscription matching (Lewis et. al., 2006). We have also begun working towards a fully composable KBN reasoner which minimises the reasoning overhead and memory footprint of the reasoning depending on the ontologies to be reasoned and application requirements.

One of the crucial tasks in the development of Siena bag extension was the implementation of the algorithm for the comparison of bags by the composite bag operator. The algorithm used is a very simple brute force algorithm that clearly illustrates the performance of the composite bag operator. It simply checks all possible arrangements of one bag with another until a matching arrangement is found, or else it returns unsuccessful. As discussed above this is not ideal. An optimisation of this algorithm would be to sort the bags (at least partially) before applying the bag comparison to the bag. This has no effect on the contents of the bags since bags are defined to be unordered, but depending on the size of the bags it can optimise the matching of bags.

## **MOTIVATIONAL CASE STUDIES**

The previously introduced KBN semantic operators support the ontological comparison and selection of ontological classes, instances and properties with regard to one another. Additionally, reasoning and inference allows ontological rules and restrictions placed within the ontology by the author, to be represented within the reasoned model. Both of these support the central and pivotal role in which the ontology takes with regard to KBN operation. The ontology can be seen as the defining structure on which semantic publications and

subscriptions are formed, and the use of ontologies differentiates KBN from CBN technologies, where a KBN is an extension to a CBN implementation. The application domain in which a KBN operates dictates and defines how the ontology is formed, all of which is central to a KBN's applicability to the chosen application domain. Here we introduce some of the factors which illustrate the technical considerations required when evaluating the applicability of a KBN in certain types of application domains. We then introduce application domains that may benefit from a KBN deployment.

Based on a synthetic benchmark for evaluating a KBN (Keeney et. al., 2006a), a number of factors need to be established to analyse the applicability of a KBN deployment in an application domain. These include: **1) Publication to Subscription ratio:** Subscriptions are stored in a hierarchical manner in the routers subscription tables, where subscription tables where subscriptions are paired with interested subscribers. Once a publication has been received at a broker/router then all subscriptions within that router must be searched for any possible matches. Therefore the performance of a knowledge-based (or content-based) network is highly dependent on the size and organisation of subscription tables in its brokers. The Siena subscription merging/matching algorithm, extended in the presented KBN implementation, is focused on the efficient partial ordering of subscriptions to maximise subscription coverage to minimise the number of subscriptions which must be checked against each publication. This creates a large overhead when adding or removing subscriptions, but is optimised for scenarios operating in a publication heavy manner. Therefore a high publication to subscription ratio would be of benefit for the presented KBN implementation. Other KBN implementations that do not expend as much effort optimising their subscription table would perform better where there are many more subscriptions than publications. **2) The Steady source of "live" Publication Data:** Having established that a high publication rate combined with a small subscription table is the preferred operational conditions for a KBN based on the Siena CBN, the next consideration surrounds the source of the publications themselves. Publications need to be sourced from data which is being rapidly updated. Looking to a generic stock market example it is easy to see that the data relating to a stock's price changes per-second whilst the markets are open (continuously with regard to currency). This level of change is important within a KBN and indeed any event-based system. Most pub/sub systems, including Siena, operate in a non-caching manner, i.e. once a publication has been checked against a routers subscription table it will not be checked again, and if a client was not subscribed to receive that publication there is no possibility of receiving it until it occurs again, if it ever does occur again. This non-caching operation prescribes that publication occur often and change rapidly. **3) Subscription Churn, un-subscribes:** Within a CBN/KBN, subscriptions can (at the users request) be followed by un-subscriptions. This removes all traces of the subscription from all subscription tables across the broker network. Using the Stock quote analogy it is easy to see that as a stock broker decides to purchase a specific stock a subscription is made for that stock, once trading is complete, an unsubscribe removes their initial subscription. This "churn" forces the brokers to apply cleanup mechanisms to their subscription/routing tables to remove stale information to speed-up the subscription matching mechanism as fewer subscriptions are searched for a possible match to incoming publications. This mechanism is however dependent upon a source of subscribing clients with dynamic and shifting subscription interests. In the Siena CBN subscriptions are aggregated together so that only the most general of subscriptions are routed between brokers, thereby minimising the size of each brokers subscription tables. This means however that when a general subscription is unsubscribed it may have a serious knock-on effect throughout the network as subscriptions that were previously covered by that subscription now need to be sent to neighbouring brokers. In scenarios with high subscription rates and substantial subscription churn this can seriously undermine the operation of the broker network as subscription table optimisation messages propagate. In such scenarios a KBN implementation that is not based on the Siena CBN would perform better. **4) Full Utilisation of Ontological operators:** to be able to fully demonstrate and exploit the range of ontological operators it is important to understand the types to which those operators can be applied. Typically CBN operators, as previously outlined, operate across Strings, Integers, Doubles, Floats, Booleans and Byte arrays.

The introduction of the ontological operators and types extends the operator set and type set over which publications and subscriptions can be formed. This additional semantic descriptiveness is only as useful as the percentage of messages within the network that use the new semantically rich message format. To demonstrate this it is necessary for messages to be formed from a rich semantic knowledge base, defined in an ontology, which can be used throughout the network and which matches the domain in which the network is operating. Full utilisation of ontological operators will only become viable once in a domain which supports full ontological messages. An important characteristic of the presented KBN implementation is that the new types and operators add no additional runtime overhead if they are unused. In such cases the KBN implementation performs identically to the Siena implementation from which it was extended. **5) Availability of usage statistics:** in a hypothetical world, the chosen application domain would be associated with a publically accessible set of usage statistics which would be able to be used for simulation and evaluation of the chosen KBN implementation. Simulations will generally occur in a concentrated time frame, e.g., simulating up to a year's worth of usage in a period of several hours or days. In searching for the ideal KBN application domain it must also provide a set of data which will allow the various rates within the network to be set based on documentable data. This data is often difficult to obtain but can, with extended research, be found or estimated. Such data can then be used to demonstrate the feasibility and utility of a KBN and allow similar systems to be evaluated, compared, and improved in an objective and scientific manner.

## **Decentralised Semantic Service Discovery**

A service-orientated architecture provides the opportunity to compose services from a number of elementary services operated by various organisations across various sites. The discovery and availability of these services are currently limited to centralised registries and compositions. The typical web service scenario consists of three parts: the first being the service provider which creates and publishes the web service, the second part involves the service brokers which maintain and manage a registry of published services and aid in their discovery, and finally the service consumers, which search the service brokers' registries for a service to fulfil their requirements. Most importantly, service discovery is reliant upon the searching of centralised repositories recording the various offerings provided by various services.

With the use of Knowledge-based Networking the dynamic and decentralised discovery of semantic services has been developed and implemented, as documented by Roblek (Roblek, 2006). The Decentralised Semantic Service Discovery system exploits ontologically based descriptions of Web services using OWL-S (Martin et. al., 2004) in order to provide an effective, efficient and distributed rendezvous of loosely coupled service providers and consumers. Participating services are described based on required capabilities in terms of inputs, outputs, preconditions and effects. Central to the detailed process of composing a composite service involves the use of constituent services where the outputs of one service correspond to the inputs of the next service, and the preconditions required by each service can be satisfied by the resulting effects of the next service in the workflow. This combines to provide a composite service which performs the tasks required given the available inputs and provides the required outputs so together the constituent services achieve the goal of the composite service. The entire process of discovering and orchestrating services is explained in detail by Roblek (Roblek, 2006). However here we focus on the matching of semantic service inputs and outputs, as explained below.

In this application domain, distributed services can announce their presence with a KBN notification. The core of this notification includes the description of the service inputs, as a bag of semantic classes, and the service outputs also represented as a bag of semantic classes. For each required service, a KBN subscription is created. The subscription uses the previously discussed composite bag operators to search for and discover compatible services or sub-services. With the routing of semantically enhanced messages being central in each KBN broker this allows for the routing of service notifications from service providers to consumers. If the bag of

inputs required by an available service is a sub-bag of the bag of available inputs when the superclass suboperator is applied ( $AdvertisedServiceInputs \subseteq_{LESSSPEC} AvailableInputs$ ), and the bag of outputs from an available service is a super-bag of the bag of the required outputs when the subclass operator is applied ( $AdvertisedServiceOutputs \supseteq_{MORESPEC} RequiredOutputs$ ), then the available service's interface is appropriate. If this is encoded as a KBN subscription, allowing for any additional semantic or content-based subscription filters to be also included in the subscription, then the KBN can act as a decentralised service discovery platform. For more details on how the other aspects of discovering, orchestrating and choreographing services is achieved, please refer to (Roblek, 2006).

Recently a similar approach was taken with the PADRES pub/sub system for the decentralised discovery and composition of services (Songlin et. al., 2008). This system maps service interface specifications to content-based pub/sub messages and is based on type-matching outputs of one service to the inputs of the next. This is, however, restricted to non-semantic services and so takes no account of the behaviour or semantics of the services being composed or the semantics of their inputs or outputs, and so the end-to-end behaviour or semantics of the composed process cannot be clearly established.

## **Distributed Correlation of Faults in a Managed Network**

Increasingly there is a demand for more scalable fault management schemes to cope with the ever increasing growth and complexity of modern networks. However, traditional fault management approaches typically involve rigid and inflexible hierarchical manager/agent topologies and rely upon significant human analysis and intervention, both of which exhibit difficulties as scale and complexity increases. Our distributed correlation scheme, designed and implemented by Tai (Tai, 2007; Tai et. al., 2008) distributes correlation tasks amongst an entire network of fault agents, where each agent takes a role in part of the correlation. These distributed agents are arranged so that low level correlators provide sub correlation results for higher level correlation agents, and the whole correlation task for the managed network can then be performed hierarchically.

Event information, correlation rules and the event correlation graphs are all represented in this scheme as ontologies. The use of an ontological representation not only enables these elements to be easily changed, but also (through reasoning) provides an opportunity for self configuration of the fault correlation system itself to be achieved automatically in reaction to context changes. We have also published numerous works describing the benefits of using semantic mark-up in the area of network fault management (Keeney et. al., 2006b; Keeney et. al., 2007a; Lewis et. al., 2005)

In one part of the work described by Tai (Tai et. al., 2008; Tai, 2007) we arranged high-level and low level events in a hierarchical manner according to a "caused\_by" relationship, where low level events cause high-level events. This was then codified using the ontological subclass/ superclass relationship. A correlation agent would then subscribe to all events at a certain level or all events that could cause a specific event using the semantic MORESPEC operator. If the agent was interested in a combination of events then it can subscribe to a flexible bag of causing events that may have occurred together. Once an agent discovers or calculates a correlation it announces this as a higher-level event using a KBN notification. (By including a bag of information about what triggered this correlation a top level agent could then perform root-cause analysis of what caused a top-level fault!).

However, we found that this mapping of caused-by relationships may not be easily map-able to a subclass/superclass relationship, and the use of this relationship to codify a "caused-by" relationship was breaking the semantics of the concept hierarchy. It was this, combined with several other factors, that prompted us to develop the generic ONTPROPERTY operator, where the "caused-by" relationship and similar relationships could be codified directly as ontological object properties without rearranging the natural hierarchy

of event types. Therefore agent subscription(s) would then match interesting events according to this causes/caused-by ontological property, (*FaultInstance* ONTPROPERTY<sub>CAUSED</sub>BY *SubFault*). This could then be easily expanded to make use of the bag extension.

There is neither tight coupling between the network of managed elements nor specific correlation agents due to the usage of the semantic publish/subscribe middleware. All events, including raw fault events are pushed into the fault correlation network, but if no agent is interested in that event then the event is quenched immediately. If an agent is interested in the event then it is routed to that agent. If there are no events in the network the correlation agent takes up minimal resources. In addition, a failure in one specific correlation agent can not disable the whole fault management system, as another correlation agent can assume the correlation task of the failed agent by adjusting its subscription.

## News Feed Distribution and Subscription

Modern web users are increasingly interested in being delivered in a timely and efficient manner, content being freshly created and posted as opposed to locating existing knowledge from a multitude of sources. A prime example of this being the subscription to a blog as a source of information as opposed to a leading encyclopaedia. The interval in-between which information on the internet is posted is becoming more and more important with regard to the content of the posting. Only the most up-to-date and relevant information is required by the modern user, e.g. blog postings rapidly fade in importance as time passes. The internet community has responded to this with the wide-spread adoption of RSS, which allows subscribers to be notified immediately of changes in a publisher's content. News-based RSS feeds have emerged, with blog subscriptions, as a cornerstone of the Web 2.0 movement. This system still relies on users actively seeking feeds which they are interested in, or being delivered information on feeds they have already subscribed to. RSS does not "push" feeds to users, users search and subscribe to feeds which match their interests. Feeds aggregators aid users in the search for information, but offer only rudimentary searches on categorised and crawled feeds.

The pub/sub model provides good suitability with regard to application such as RSS. The examples in use typically are limited to a number of static events and event types, and for this reason Semantic web techniques have yet to be fully deployed in the efficient distribution of events. The work presented in (Keeney et. al., 2008b) introduces a real-world study of the distribution and subscription of multiple consumers to podcast feeds, utilising the apple podcast schema, which adds to RSS XML tags. It was found that podcast feeds and their individual postings are marked up with a small amount of metadata and that this data could be extended, naturally, to utilise the semantic extensions offered through the use of knowledge-based networking. This hidden metadata includes information such as classification, categorisation and keywords associated with the podcast (which would form the publication). The categorisation of the feeds was found to be usually drawn from a taxonomy of categories, which could be easily ported to and extended into the structure of a rich ontology.

Having encoded the scope of the podcasts categorisation ontologically, it became possible to encode the publication of a new podcast in the form of a KBN notification which contained authorship and ownership information, the URL of the actual podcast post, an attribute containing a bag of ontological classes as subject categories and a bag of keywords. These publications were passed to the KBN brokers in line with usage statistics collected and documented in (Keeney et. al., 2008b). Having established the source of publications, consumers of events could then receive events based on a KBN subscription. Among other filters the subscriber would specify a bag of zero or more required keywords and a bag of zero or more categories. Firstly the bag of keywords in the event notification was formed around a simple superbag of the keywords in the subscription (*EventKeywords*  $\supseteq$  *RequiredKeywords*). Subsequently if the subscription contained a bag of *required* categories then the bag of categories in the event notification should be a *superbag* of the bag of categories requested, or

equivalent. If the subscription contained a bag of *suggested* categories then the bag of categories in the event notification should be a *subbag* of the bag of categories requested or equivalent. Since the categories were arranged taxonomically, the subscription should match equivalent categories and their sub-categories, so the subscription used the MORESPEC suboperator ( $EventCategories \supseteq_{MORESPEC} RequiredCategories$ ) or ( $EventCategories \subseteq_{MORESPEC} SuggestedCategories$ ). For more details on this scenario, and a detailed evaluation of the performance of the KBN in this usage scenario, refer to (Keeney et. al., 2008b).

Similar approaches for RSS feed aggregation have also been proposed, both semantic based, e.g. S-ToPSS (Petrovic et. al., 2005) and non-semantic based, e.g. (Roitman et al, 2008). Both systems attempt to maintain the interests (profile) of a user as a set of shifting subscriptions, whereby RSS feeds are polled, aggregated and presented to the user according to their interests and/or semantics. Both works further strengthen the argument behind the need for a knowledge-discovery mechanism, as the size of searched content increases and the need for a dynamic mechanism for the management of knowledge further becomes a necessity.

## Context Distribution

Pervasive computing promises to make available a vast volume of context messages from environmental sensors embedded in the fabric of everyday life, reporting on user location sound levels and temperature changes, to name but a few. Any scalable context delivery system must ensure therefore the accurate delivery of context events to the consumers that require them. However, the wide range of sensors and sensed information, and the mobility of consuming clients, will present a level of heterogeneity that prevents consumers accurately forming queries to match possibly unknown forms of relevant context events. As context-aware systems become more widespread and more mobile there is an increasing need for a common distributed event platform for gathering context information and delivering to context-aware applications. However, most pub/sub systems require agreements on message types between the developers of producer and consumer applications. This places severe restrictions on the heterogeneity and dynamism of client applications. Here we see an ideal application potential of Knowledge-based Networks for the filtered dissemination of context over a large loosely coupled network of distributed heterogeneous agents, while removing the need to bind explicitly to all of the potential sources of that context. The likely heterogeneity across the body of context information can be addressed using runtime reasoning over ontology-based context models. A KBN based solely on semantically enhanced messages and corresponding expressive and flexible queries is far more flexible, open and reusable to new applications. For this reason we foresee the application of, and have already applied, KBNs in numerous context-aware scenarios (Keeney et. al., 2006a; Keeney et. al., 2006b; Keeney et. al., 2007a).

## Discussion of Case Studies

As described in (Raiciu et. al., 2006) no single implementation or configuration of a content-based (or knowledge-based) pub/sub system will perform well for all application scenarios. Different scenarios require different configurations depending on a number of factors: subscription language expressiveness, publisher bandwidth, latency tolerances, clustering capabilities, the number of subscribers and publishers, the frequency of publications and subscriptions etc.. When extended to knowledge-based networking the main additional factors centre on the amount, complexity, expressiveness and spread of semantics across the network of brokers, publishers and subscribers, just as the addition of semantics marks a KBN different from a CBN. A full list of the content-based and knowledge-based factors that characterise a KBN deployment is presented in (Keeney et. al., 2006a).

Just as a single CBN implementation or configuration will struggle to perform optimally for all applications, the KBN implementation presented is not intended for all applications and deployments since it inherits the characteristics of the Siena CBN upon which it is implemented. The motivational case studies presented are

intended to illustrate the contributions and relative advantages of exploiting semantics deep in the pub/sub network. It is envisioned that as more knowledge-based networking implementations (in addition to those surveyed earlier) begin to appear, this will provide a more diverse set of systems to cover such a diverse set of applications and deployments.

The performance and appropriateness of a KBN deployment for decentralised semantic service discovery is very similar to that of a CBN for non-semantic web services. Semantic web services are generally well marked up in terms of the semantics of the behaviour of the service, the semantics of its inputs, outputs, preconditions and effects (Martin et. al., 2004). A KBN provides a mechanism for service consumers to form expressive queries for services, not just in terms of semantics but also in terms of non-semantic metadata describing the services. It is likely that the semantic operators used in such subscriptions will be relatively restricted to the ontological concept subsumption operators rather than instance or property operators. As event notifications are generally not cached in a pub/sub system, the lack of a registry means that publishers must regularly announce themselves, where the frequency of these announcements is dependent on the requirements of the consumers to find services in a timely manner. This means that the number of publications grows quickly, and so publications that do not match a subscription must be quenched as close to the source as possible. This necessitates that subscriptions are widely distributed in the network. At any time some popular services will likely have many subscribers while most will likely have no subscribers. Subscriptions will generally be quite short lived as the service consumer will remove their subscription after they have found an appropriate service, so the KBN configuration chosen must be tolerant of substantial subscription churn.

The performance and appropriateness of a KBN deployment for distributed fault correlation in network management depends largely on the size of the network being managed and the services it supports. When faults are signalled as publications, it is necessary that the pub/sub mechanism can support huge numbers of events when an “event storm” occurs with possibly hundreds of events per second in irregular bursts. In this scenario there will be relatively few subscribers, but the total number of subscriptions will depend on the faults that the network administrator anticipates and for which correlation tasks have been encoded as subscriptions. It is still expected that the number of publications (in bursts) will greatly outnumber the number of subscriptions. It is also expected that the subscriptions will be long-lived, unsubscriptions will be rare and so subscription churn will be low. The advantage of a KBN over other types of pub/sub mechanisms is directly related to the use of semantics. While low-level events from network elements will have relatively little semantics (aside from statically encoded metadata), the higher-level events will become increasingly semantically rich as correlation tasks are performed. Therefore, only rudimentary operators are used at the lowest levels, but with more expressive operators used higher up to exploit the flexibility of the subscription mechanism for use in the complicated correlation process.

In a news feed distribution scenario scalability becomes the major concern. As users of the web are increasingly interested in tracking the appearance of new postings rather than locating existing knowledge the use of a pub/sub mechanism is ideal for this scenario. Some feeds have large numbers of subscribers, where many feeds have very few subscribers. Based on information from (Keeney et. al., 2008b) subscriptions are generally long-lived, with considerable semantic drift in new subscriptions but subscription churn is small. The number of publishers are moderately large, but the number of publications each publisher produces is relatively few. In contrast there are more subscribers than publishers so it is necessary to efficiently aggregate, and perhaps cluster subscriptions across the network. Using a CBN, user subscriptions are limited to simple syntactic matches (typically integers, strings and Booleans) but with the popularity of semantic-based metadata within the Web 2.0 community more expressive and flexible semantic subscriptions are required. A problem arises from the vast and shifting semantic knowledge-base required for this scenario. It is necessary that any large scale KBN implementation can partition or semantically cluster the network to localise semantic drift and handle the large semantic spread across the network.

The context distribution domain is much more difficult to quantify. The number of subscribers and publisher, the number of messages, subscription churns, the use of semantics, the expressiveness of the subscriptions etc. are all impossible to quantify and are completely application dependent. For example, large deployments of sensors will likely produce a vast number of publications, but with relatively little semantics in each. There will probably be few subscribers interested in such low-level information. In contrast, context aggregators, application level context producers, sophisticated sensors, and environment controllers will publish few events but they will be contain rich semantics and may be of interest to a larger set of subscribers. There will be little semantic drift as the semantics of the context will not change, even if the context values may change rapidly. Subscriptions for low level event will probably be long lived, but high level semantic subscription may be shorter lived as the subscriber discovers the context they were interested in, but overall there will be little subscription churn and subscription patterns will be relatively predictable.

### **Additional proposed scenarios**

**Academic Calls for Papers** are unique in their global interest combined with regular churn of both subscribers and publishers, in the form of conference organisers and delegates. Semantically rich messages in the form of contribution calls, can focus mainly on the semantically rich message format as used within a KBN as opposed to static type comparison used within most Content-Based Networks. The relationship between areas in computer science, the conference location, the types of papers being solicited and the dates and duration of the conference can all be represented within a rich and expressive CFP ontology, from which both publication and subscription can be formed.

**Commodities and Currencies** provide two interesting and popular areas in which KBNs could successfully operate, with the demand for timely and accurate information requiring such a semantically enhanced pub/sub system. Commodities can be classified as Heating Oil or Light Crude, Silver or Copper and Live Cattle or Lean Hogs, to name just a few. These commodities have a current price, percentage change, high/low and an update period. Currencies are very much the same offering cross rates (between various currencies) and percentage changes, all providing a vast set of data. Much of this data is ideally suited for semantic mark-up exploiting ontologies representing the change and relationship between commodities and the market itself.

**Environmental Controls / Building Management:** in the proposed ubiquitous world, environmentally sensitive buildings will produce vast amounts of data on which time sensitive decisions will need to be made. Where the semantically enhanced pub/sub paradigm is best operational is within an environment in which rapid changes in the form of publications are matched to pre-existing sorted and stored subscriptions. The distributed filtering of vast amounts of semantically enriched data produced by environmental controls, and ability to perform long-running semantic queries over the data, is an ideal application scenario for a KBN.

**Hospital equipment monitoring critically ill patients:** there are many areas in hospital care and medicine in which the consumer and producer would benefit from the loose coupling offered through pub/sub systems. Patient care is something in which advances in computing have shown to have a positive effect with regard to day-to-day care. If each patient, or the machines monitoring or treating the patient, were seen as publishers then this would allow a healthcare professional to subscribe to alerts when specific criteria are met. It is easy to see the obvious benefits of this scenario, however with such critical data it is crucial that this data would be delivered in an assured and timely manner, an area in which further research is required.

### **CONCLUSION AND FURTHER WORK**

This paper describes and discusses the implementation of two novel extensions to the Siena Content-based Network to extend it to become a new class of Knowledge-based Networking. One extension provides

ontological concepts as an additional message attribute type, onto which subsumption relationships, equivalence, type queries and arbitrary ontological relationships can be applied. The second extension provides a bag type to be used allowing bag equivalence, sub-bag and super-bag relationships to be used directly in the subscription filters, composed with any of the Siena subscription operators or the ontological operators previously mentioned. This research has only just begun to explore applications for the expressiveness of the knowledge-based networking. As presented in the motivational case studies above, ongoing research by the authors is focussing on how our KBN implementation can be applied across a wide selection of application areas, including:

- Decentralised semantic service discovery (Roblek, 2006)
- Discovery and change notification of policies between federated communication service providers
- Sensor readings in a multi-domain heterogeneous ubiquitous computing application
- RSS extended with semantic mark-up in Web 2.0/Semantic Web (Keeney et. al., 2008b)
- Semantically rich notifications from heterogeneous network elements in Operational Support Systems (Lewis et. al., 2005; Keeney et. al., 2006b; Keeney et. al., 2007a)
- Distributed fault correlation using semantically rich notifications (Tai, 2007; Tai et. al., 2008)
- Semantically rich notifications about changes in financial markets
- Semantic routing of multimedia (MPEG) stream with semantic meta-data

One of the main questions that surround the use of ontologies deep in the network at the routing layer remains the evaluation of the resulting performance overhead. Previous small scale studies in this area (Lewis et. al., 2006; Keeney et. al., 2006b; Keeney et. al., 2007a; Keeney et. al., 2008a) show a definite performance penalty but this may be acceptable when offset against the increased flexibility and expressiveness of the KBN subscription mechanism. Further research is required to evaluate how the performance of “off-the-shelf” ontology tools will affect the scalability of KBNs within larger scales. These results point to the potential importance of semantic clustering for efficient network and performance scalability. Ongoing work is also focussing on extending the Knowledge-Based Network to incorporate semantic-based clustering. This work aims to provide a network environment in which routing nodes, publishers and subscribers are clustered based on their semantic footprint and interests (Keeney et. al., 2008b). The benefits of this are threefold: Firstly, this reduces the processing time involved in making routing decisions based on the messages content. Its take fewer hops to get from source to destination, as these are already closely linked based on the likelihood of there being a match between the two. Secondly, this allows for natural grouping of likeminded publishers and subscribers as seen in traditional web forums / newsgroups. Thirdly, it allows certain areas of the network to have specialised sub/super ontologies which do not need to contain the semantics of the whole network. This means that the knowledge base sizes can reduced and the knowledge base updates can be localised. This cluster-based approach to pub/sub networks turns the normal user-based search paradigm full circle as network data is passed from node to node towards those who are most likely to be interested in the data as opposed to those users searching out that same data. Our initial work clusters were statically designed and operated (Keeney et. al., 2008b). In this sense nodes were assigned to clusters without the possibility of changing clusters once they have joined, in a manner similar to the approach taken in (Baldoni et. al., 2007). This initial clustering method demonstrated how even inflexible and static clustering can have a substantial positive effect on overall performance. However, we expect that any practical system will need to adapt its clustering to reflect the constantly changing profile of semantics being sent and subscribed to via a KBN, thus creating a network environment in which messages are passed from node-to-node, cluster-to-cluster based not on the data’s destination but based on the message’s semantic data. Recently completed work focussed on allowing users and brokers to join and leave clusters dynamically and independently. Clusters can then be seen as organic structures in which users and brokers join and leave as their own personal interests drift, grow, reform and refine. Current work is also focusing on integrating policy-based cluster management for a KBN to supports sophisticated clustering schemes. This will support overlapping clusters and hierarchies of clusters under separate administrative control (Lewis et. al., 2006). In addition, the

effect of semantic interoperability in node matching functions and in inter-cluster communications is being assessed (GuoS et. al., 2007; GuoS et. al., 2008). This requires evaluation of different schemes for injecting newly discovered semantic interoperability mappings into the ontological corpus held by KBN routers, as well as how these mappings are shared between routers. Work is ongoing to build on these initial evaluations (GuoS et. al., 2007; GuoS et. al., 2008) to design and implement a flexible mapping strategy management framework. Work is also ongoing to investigate how mappings can be dynamically distributed around the network as the knowledge bases of clients joining and leaving the network affect the spread of knowledge across the network. It is foreseen that a KBN itself would be ideal for such a distribution mechanism.

## REFERENCES

- (Baldoni et. al., 2007) Baldoni, R., Beraldi, R., Quema, V., Querzoni, L., Tucci-Piergiovanni, S., (2007), "TERA: topic-based event routing for peer-to-peer architectures," In Proceedings of Distributed event-based systems, (DEBS2007) New York, NY, USA, 2007.
- (Borsje et. al., 2008) Borsje, J., Levering, L., Frasinca, F., (2008), "Hermes: a Semantic Web Based News Decision Support System," in *The 23rd Annual ACM Symposium on Applied Computing* Fortaleza, Ceará, Brazil, March 16 - 20, 2008.
- (Burcea et. al., 2003) Burcea, I., Petrovic, M., Jacobsen, H-A., (2003), "I know what you mean: Semantic Issues in Internet-scale Publish/Subscribe Systems," In Proceedings of the International Workshop on Semantic Web and Databases (SWDB03), Berlin, Germany, 2003.
- (Cai et. al., 2004) Cai, M., Frank, M., (2004), "RDFPeers: A scalable distributed RDF repository based on a structured peer-to-peer network", In Proceedings of WWW conference, May 2004, New York, USA.
- (Carroll et. al., 2004) Carroll, J., Dickinson, I., Dollin, C., (2004), "Jena: Implementing the Semantic Web Recommendations", In Proceedings of World Wide Web Conference 2004, 17-22 May 2004, New York, NY, USA. <http://jena.sourceforge.net/>.
- (Carzaniga et. al., 2008) Carzaniga, A., "Siena - Software", (2008), <http://www.inf.unisi.ch/carzaniga/siena/software/index.html>
- (Carzaniga et. al., 2001) Carzaniga, A., Rosenblum, D. S., and Wolf, A. L. (2001). Design and Evaluation of a Wide-Area Event Notification Service. *ACM Transactions on Computer Systems*, 19(3).
- (Chan et. al., 2006) Chan, C.Y., Ni, Y., (2006), "Content-based Dissemination of Fragmented XML Data". In Proceedings of International Conference on Distributed Computing Systems, ICDCS 2006: 44
- (Chirita et. al., 2004) Chirita, P.-A., Idreos, S., Koubarakis, M., Nejdl, W., (2006), "Publish/Subscribe for RDF-Based P2P Networks," In Proceedings of the 1st European Semantic Web Symposium (ESWS 2004), Heraklion, Greece, May 10-12, 2004.
- (Christian et. al., 2007) Christian, H.-W., James, H., (2007), "Toward expressive syndication on the web," In Proceedings of the 16th international conference on World Wide Web, Banff, Alberta, Canada: ACM, 2007
- (Cilia et. al., 2003) Cilia, M., Bornhövd, C., Buchmann, A. P., (2003), "CREAM: An Infrastructure for Distributed, Heterogeneous Event-Based Applications". *CoopIS 2003*, Catania, Sicily, Italy,
- (Clark et. al., 1999) Clark J., DeRose, S., (1999), "Xml path language (xpath)". <http://www.w3.org/TR/xpath>.
- (Diao et. al., 2003) Diao, Y., Altinel, M., Franklin, M.J., Zhang, H., Fischer, P., (2003), "Path sharing and predicate evaluation for high-performance xml filtering". *ACM Transactions on Database Systems (TODS)*, 28(4):467–516, 2003.

- (Eugster et. al., 2003) Eugster, P., Felber, P., Kenmarrec, A.M., and Guerrout, R., (2003), "The many faces of publish/subscribe". ACM Computing Surveys (CSUR), Vol. 35, Issue 2, (June 2003), 2003.
- (Grossnickle et. al., 2005) Grossnickle, J., Board, T., Pickens, B., Belmont, M., (2005), "RSS - Crossing Into the Mainstream" Yahoo! & Ipsos Insight Oct 2005.
- (GuoS et. al., 2008) Guo, S., Keeney, J., O'Sullivan, D., Lewis, D., (2008), "Coping with Diverse Semantic Models when Routing Ubiquitous Computing Information", The Workshop on Managing Ubiquitous Communications and Services (MUCS2008) at NOMS 2008, Salvador, Bahia, Brazil, 7-11 April 2008.
- (GuoS et. al., 2007) Guo, S., Keeney, J., O'Sullivan, D., Lewis, D., (2007), "Adaptive Semantic Interoperability Strategies for Knowledge Based Networking", In Proceedings of International Workshop on Scalable Semantic Web Knowledge Base Systems (SSWS '07) at OTM 2007, Vilamoura, Portugal, 27-29 November 2007.
- (GuoY et. al., 2004) Guo Y., Heflin J., Pan Z., (2004), "An Evaluation of Knowledge Base Systems for Large OWL Datasets", Technical Report, CSE department, Leigh University, 2004.
- (GuoY et. al., 2005) Guo Y., Heflin J., (2005), "LUBM: A Benchmark for OWL Knowledge Base Systems", Journal of Web Semantics, Vol. 3 Issue 2, 2005.
- (Gupta et. al., 2003) Gupta A., Suciu. D., (2003), "Stream processing of xpath queries with predicates". In Proceedings of 2003 ACM SIGMOD Intl conference on Management of data pages 419–430, 2003.
- (Haarslav et. al., 2001) Haarslev, V., Moller, R. (2001), "RACER System Description", In Proceedings of IJCAR 2001, volume 2083 of LNAI, 701–706. Siena, Italy, Springer.
- (Jennings et. al., 2007) Jennings, B., van der Meer, S., Balasubramaniam, S., Botvich, D., O'Foghlu, M., Donnelly, W., Strassner, J., "Towards Autonomic Management of Communications Networks", IEEE Communications Magazine, 45(10), October 2007
- (Keeney et. al., 2006a) Keeney, J., Lewis, D., O'Sullivan, D., (2006), "Benchmarking Knowledge-based Context Delivery Systems", In Proceedings of ICAS06, Silicon Valley, USA, July 19-21, 2006.
- (Keeney et. al., 2006b) Keeney, J., Lewis, D., O'Sullivan, D., Roelens, A., Wade, V., Boran, A., Richardson, R., (2006), "Runtime Semantic Interoperability for Gathering Ontology-based Network Context", Network Operations and Management Symposium (NOMS 2006), Canada. April 2006.
- (Keeney et. al., 2007a) Keeney, J., Lewis, D., O'Sullivan, D., (2007), "Ontological Semantics for Distributing Contextual Knowledge in Highly Distributed Autonomic Systems", Journal of Network and System Management, Vol 15, March 2007
- (Keeney et. al., 2008a) Keeney, J., Roblek, D., Jones, D., Lewis, O'Sullivan, D., (2008), "Extending Siena to support more expressive and flexible subscriptions," in The 2nd International Conference on Distributed Event-Based Systems (DEBS 2008), Rome, Italy, 2008.
- (Keeney et. al., 2008b) Keeney, J., Jones, D., Roblek, D., Lewis, D., O'Sullivan, D., (2008), "Knowledge-based Semantic Clustering," In Proceedings of ACM Symposium on Applied Computing, Fortaleza, Brazil, 2008.
- (Lewis et. al., 2005) Lewis, D., O'Sullivan, D., Power, R., Keeney, J., (2005), "Semantic Interoperability for an Autonomic Knowledge Delivery Service", In Proceedings of Workshop on Autonomic Communication (WAC 2005), Vouliagmeni, Athens, Greece. October 2005.
- (Lewis et. al., 2006) Lewis, D., Keeney, J., O'Sullivan, D., Guo, S., (2006), "Towards a Managed Extensible Control Plane for Knowledge-Based Networking", Distributed Systems: Operations and Management Large Scale Management, (DSOM 2006), at Manweek 2006, Dublin, Ireland, 23-25 October 2006

- (Li et. al., 2004) Li, H., Jiang, G., (2004), "Semantic Message Oriented Middleware for Publish/Subscribe Networks", In Proceedings of SPIE, Volume 5403, pp 124-133, 2004
- (Loser et. al., 2003) Loser, A., Naumann, F., Siberski, W., Nejd, W., Thaden, U., (2003), "Semantic overlay clusters within super-peer networks", In Proceedings of Workshop on Databases, Information Systems and Peer-to-Peer Computing in Conjunction with the VLDB 03
- (Lynch et. al., 2006) Lynch, D., Keeney, J., Lewis, D., O'Sullivan, D., (2006), "A Proactive Approach to Semantically Oriented Service Discovery". Innovations in Web Infrastructure (IWI 2006). at World-Wide Web Conf., Edinburgh, Scotland. May 2006.
- (Martin et. al., 2004) Martin, D., Burstein, M., Hobbs, J., Lassila, O., McDermott, D., McIlraith, S., Narayanan, S., Paolucci, M., Parsia, B., Payne, T., Sirin, E., Srinivasan, N., Sycara, K., (2004), "OWL-S: Semantic Markup for Web Services", W3C Member Submission 22 November 2004
- (Meier et. al., 2005) Meier, R., Cahill, V., (2005), "Taxonomy of Distributed Event-Based Programming Systems", The Computer Journal, vol 48, no 5, pp 602-626, 2005
- (Motik et. al., 2006) Motik, B., Sattler, U., (2006), "Practical DL Reasoning over Large ABoxes with KAON2", available at <http://kaon2.semanticweb.org/>
- (Mühl et. al., 2006) Mühl, G., Fiege, F., Pietzuch, P., (2006), "Distributed Event-Based Systems". Springer-Verlag, 2006.
- (owl, 2004) OWL, (2004), "W3C Recommendation: OWL Web Ontology Language Overview", "<http://www.w3.org/TR/owl-features/>," Accessed June 2008.
- (Pan, 2005) Pan Z., (2005), "Benchmarking DL Reasoners Using Realistic Ontologies", In Proceedings of Intl workshop on OWL: Experience and Directions (OWL-ED2005). Galway, Ireland. 2005
- (Parsia et. al., 2004) Parsia, B., Sirin, E., (200V), "Pellet: An OWL-DL Reasoner", Poster at ISWC 2004, Hiroshima, Japan, 2004.
- (pellet, 2003) "Pellet Performance", (200V), <http://www.mindswap.org/2003/pellet/performance.shtml>
- (Peitzuch et. al., 2002) Pietzuch, P., Bacon, J., (2002), "Hermes: A Distributed Event-Based Middleware Architecture," In Proceedings of International Conference on Distributed Computing Systems, 2002.
- (Peitzuch et. al., 2003) Pietzuch, P., Bacon, J., (2003), "Peer-to-Peer Overlay Broker Networks in an Event-Based Middleware". Distributed Event-Based Systems (DEBS'03). In Proceedings of the ACM SIGMOD/PODS Conference, San Diego, CA, June 2003.
- (Petrovic et. al., 2003) Petrovic, M., Burcea, I., Jacobsen, H. A., (2003), "S-ToPSS: semantic Toronto publish/subscribe system," In Proceedings of the 29th international conference on Very large data bases (VLDB03), Berlin, Germany, 2003.
- (Petrovic et. al., 2005) Petrovic, M., Liu, H., Jacobsen, H.-A., "CMS-ToPSS: Efficient Dissemination of RSS Documents". In Proceedings of 31st International Conference on Very Large Data Bases (VLDB). September 2005.
- (Raiciu et. al., 2006) Raiciu, C., Rosenblum, D. S., Handley, M. "Revisiting Content-Based Publish/Subscribe". In Proceedings of 26th IEEE international Conference on Distributed Computing Systems Workshop, 4-7 July 2006.
- (Roblek, 2006) Roblek, D. (2006), "Decentralized Discovery and Execution for Composite Semantic Web Services", M.Sc. Thesis, Computer Science, Trinity College Dublin, Ireland, 2006.

- (Roitman et. al., 2008) Roitman, H., Carmel, D., Yom-Tov, E., (2008), “Maintaining dynamic channel profiles on the web”. In the 34th Conference on Very Large Data Bases (VLDB 2008), Auckland, New-Zealand. 2008.
- (Segal et. al., 2000) Segall, B., Arnold, D., Boot, J., Henderson, M., Phelps, T., (2000), “Content-Based Routing in Elvin4”, In Proceedings of AUUG2K, Canberra 2000.
- (Schlosser et. al., 2002) Schlosser, M. Sintek, M., Decker, S., Nejd, W., “HyperCuP — Hypercubes, Ontologies and Efficient Search on P2P Networks”, In Proceedings on the International Workshop on Agents and Peer-to-Peer-Systems, Bologna, Italy, 2002, Springer.
- (Skovronski et. al., 2006) Skovronski, J., Chiu, K., (2006), “Ontology Based Publish Subscribe Framework”. In Proceedings of International Conference on Information Integration and Web-based Applications Services, 4-6 December 2006, Yogyakarta, Indonesia.
- (Songlin et. al., 2008) Songlin, H., Muthusamy, V., Li, G., Jacobsen, H.-A., “Distributed Automatic Service Composition in Large-Scale Systems”. in The 2nd International Conference on Distributed Event-Based Systems (DEBS 2008), Rome, Italy, 2008.
- (Tai, 2007) Tai, W., (2007), “Fault Management System using Semantic Publish/Subscribe approach”, M.Sc. Thesis, Computer Science, Trinity College Dublin, Ireland, December 2007.
- (Tai et. al., 2008) Tai, W., O'Sullivan, D., Keeney, J., (2008), "Distributed Fault Correlation Scheme using a Semantic Publish/Subscribe system," In Proceedings of Network Operations and Management Symposium (NOMS 2008), Salvador, Brazil, April 2008.
- (Tempich et. al., 2004) Tempich, C., Staab, S., Wranik, A., (2004), “REMINDIN’: semantic query routing in peer-to-peer networks based on social metaphors” International World Wide Web Conference (WWW), New York, USA, 2004.
- (w3c, 2003) W3C: “The Wine Ontology” (2003), <http://www.w3.org/TR/owl-guide/wine.rdf>
- (Wang et. al., 2004) Wang, J., Jin, B., Li, J., (2004), “An ontology-based publish/subscribe system”. In Proceedings of ACM/IFIP/USENIX International Conference on Middleware, 2004.
- (Weisstein, 2002) Weisstein, E. W. (2002). “Multiset. MathWorld – Wolfram Resource”. <http://mathworld.wolfram.com/Multiset.html>
- (Zou et. al., 2004) Zou Y., Finin T., Chen H., (2004), “F-OWL: an Inference Engine for the Semantic Web”, In Proceedings of Workshop on Formal Approaches to Agent Based Systems, April 2004, MD, USA, LNCS 3228