Ontological Semantics for Distributing Contextual Knowledge in Highly Distributed Autonomic Systems

John Keeney, David Lewis, Declan O'Sullivan

Centre for Telecommunications Value-chain Research and

Knowledge and Data Engineering Group

School of Computer Science and Statistics,

Trinity College Dublin,

Dublin, Ireland.

{John.Keeney | Dave.Lewis | Declan.OSullivan }@cs.tcd.ie

Corresponding Author:
Dr. John Keeney
School of Computer Science and Statistics
Trinity College Dublin
College Green
Dublin 2, Ireland
Email: John.Keeney@cs.tcd.ie
Tel: +353-1-8968431, Fax: +353-1-6772204

ABSTRACT

Much recent research has focused on applying Autonomic Computing principles to achieve constrained self-management in adaptive systems, through self-monitoring and analysis, strategy planning, and self adjustment. However, in a highly distributed system, just monitoring current operation and context is a complex and largely unsolved problem domain. This difficulty is particularly evident in the areas of network management, pervasive computing, and autonomic communications. This paper presents a model for the filtered dissemination of semantically enriched knowledge over a large loosely coupled network of distributed heterogeneous autonomic agents, removing the need to bind explicitly to all of the potential sources of that knowledge. This paper presents an implementation of such a knowledge delivery service, which enables the efficient routing of distributed heterogeneous knowledge to, and only to, nodes that have expressed an interest in that knowledge. This gathered knowledge can then be used as the operational or context information needed to analyze to the system's behavior as part of an autonomic control loop. As a case study this paper focuses on contextual knowledge distribution for autonomic network management. A comparative evaluation of the performance of the knowledge delivery service is also provided.

KEYWORDS

Highly distributed autonomic systems, semantically enriched knowledge distribution, content based networking, semantic interoperability, knowledge delivery service.

RUNNING HEADER

Distributing Knowledge in Distributed Autonomic Systems

1. INTRODUCTION

Autonomic systems use knowledge of their operational state and operational context to self-manage, i.e. to self-configure, self-heal, self-optimize and self-protect, by monitoring state and context, planning and adapting. Though the need to self-manage was initially recognized as a challenge in dramatically reducing the operating costs of complex computing systems [1], increasingly complex networked and distributed systems are also seen as needing to self-manage. Applying autonomic approaches to networks, pervasive computing and other highly distributed systems represents particular challenges in gathering operational knowledge from across the system. Here operational network knowledge refers to operational state information about the system, and is accompanied by its meta-data, e.g., expressed as a management information model. The challenge arises because in a highly distributed autonomic system, the system elements that possess this knowledge are widely distributed, are purchased from different vendors, perform different functions, possess a wide range of knowledge meta-data and are operated by different organizations. A range of distributed artificial intelligence techniques, e.g. multi-agent systems, swarm intelligence, or cellular automata [2], are proposed for monitoring and analyzing network conditions in order to drive the planning of optimization, protection or corrective strategies [3]. These typically use explicit knowledge models at run time to dynamically discover, handle and reason over dynamically changing operational context information [4], and are predicated on a common means for expressing knowledge-rich meta-data. This is now possible due the standardized approaches to expressing ontological knowledge, e.g. the Web Ontology Language (OWL) [5], proposed by the Semantic Web initiative at the World Wide Web Consortium [6]. Ontology-based semantics support encoding and mapping between separately authored and thus heterogeneous sources of knowledge. This approach promises loose semantic coupling between autonomic applications, which is vital as new waves of applications increasingly rely on using the information and services offered by existing heterogeneous distributed applications. A key platform technology for the practical transition to autonomic management, therefore, is a service that enables the efficient delivery of distributed operational knowledge to, and only to, nodes that have expressed an interest in that knowledge for use as the context information needed to make appropriate adaptation to their behavior. The trends for pushing more operational intelligence towards each self managing distributed element to achieve more context-awareness in self-managing behavior often requires elements to gather distributed knowledge without necessarily binding explicitly to all of the potential sources of that knowledge. There has been some interest recently in developing middleware for accessing operational knowledge as the context for adaptive or autonomic systems in an ontological form [7][8]. To date, however, there has been no movement towards an inter-working consensus for these technologies or on how the knowledge required to make autonomic decisions is gathered from across a heterogeneous network, and particularly across administrative domains. We have already argued the need for an autonomic knowledge delivery service (KDS) that can inherently scale to the size of the system it supports, including to Internet scales [9]. To be a reliable medium for the dissemination of the knowledge needed by autonomic functions the implementation of a knowledge delivery service clearly needs to itself exhibit self-management.

In this paper we begin to address the challenge of establishing an Internet-scale knowledge delivery service for distributed, autonomous autonomic systems. Meeting this challenge demands we address both the extreme heterogeneity and rapid evolution of autonomic applications and context information, in combination with the need for high throughput, low-latency of messages between large, volatile populations of service clients. It should be clear however that any software-based event forwarding algorithm will struggle to match the hardware optimized performance of packet forwarding in IP routers.

We describe the introduction of ontological reasoning into a Publish-Subscribe content-based event delivery mechanism, measure its performance and discuss the implication for future event routing approaches. Basing the forwarding algorithm on today's ontological reasoners incurs a heavy computational load. We do not attempt to develop optimized reasoners for a knowledge delivery service, instead we aim to explore the performance of ontological reasoning to better understand how it can effectively deployed in a knowledge delivery service (KDS). Ultimately we believe this will guide the evolution of intelligent clustering in event routing algorithms that are cognizant of the performance profiles of existing reasoners and of the semantics being exchanged by client applications, and can thereby off-set the relatively poor forwarding algorithm performance ontology access and reasoning [10].

2. STATE OF THE ART AND RELATED WORK

The Publish-Subscribe (Pub-Sub) model for communication lies firmly in the scope of loosely-coupled, large-scale distributed systems [11]. In the domain of large scale loosely coupled distributed systems, Pub-Sub has emerged as one of the more promising communications models. The model consists of three basic elements; Subscribers, who express interest in particular information by means of a subscription language, Publishers of information, who publish information of interest, and an intermediary event notification service connecting the two. Pub-Sub event systems are considered as the basis for the proposed knowledge delivery service as they avoid close coupling between producers of events and one or more event consumers that have expressed an interest in an event type. However, such Pub-Sub systems require agreements on message types between producers and consumers that place severe restrictions on the heterogeneity and dynamism of client applications.

Pub-Sub systems that route and filter events based on matching message contents and attributes to client subscriptions rather than using the full message type, known as Content-Based

Networks (CBN), facilitates still looser coupling between producer and consumer applications. Several CBN solutions and prototypes exist, e.g. Siena [12], Elvin [13], HERMES [14], XNET [15] and Gryphon [16], however their scalability is not yet proven to Internet dimensions.

Widespread CBN deployments have been slow to emerge partly due to the difficulty in reaching a general compromise between the expressiveness of the contents of the events and subscription filters and the need of CBN nodes both to efficiently match these filters to events and to efficiently maintain hierarchical forwarding tables by aggregating new subscriptions with any existing ones that cover a superset of matching messages [17]. As a result current CBNs only support a very limited range of data types and operators (typically integers, strings, booleans), which falls well short of supporting the heterogeneity and flexibility that an autonomic knowledge delivery service requires. Selecting a more expressive language involves a difficult trade-off, since higher level features, e.g. set functions, introduce more complexity into a CBN node, and may only be of use to a subset of applications. We must aim therefore to have a CBN message and subscription language that can be expanded incrementally to meet the requirements of specific autonomic application domains without placing unnecessary overheads on the network as a whole.

A CBN based on publications and subscriptions containing semantic mark-up is potentially far more flexible, open and reusable to new applications. We call such a semantic-based CBN a Knowledge-Based Network (KBN), and we present this as the mechanism by which a knowledge delivery service can be implemented. There has been little examination of the use of ontology-based semantics in CBNs in the scientific literature. In [18], an extension to the Toronto Publish-Subscribe System (ToPSS) is described that proposes extending the event/subscription matching function of this CBN to include class equivalence, ontological sub-class and super-class relationships (i.e. subsumption) and semantic mapping based relationships, which is equivalent to the CBN extensions carried out in [19], [20] and in this paper. More significantly, however, no report of an implementation or evaluation of this proposal has yet emerged. In [19] and [9] previous versions of semantic Pub-Sub systems are presented, but they are based on a centralized Pub-Sub bus implementation, the commercially available Elvin system [13], which was thus limited in scalability. As Elvin is a closed-source system, semantic matching and semantic interoperability functions were implemented in client-based wrapper in those systems.

3. SUBSUMPTIVE EXTENSION TO THE SIENA CBN

Here we address how operational knowledge needed to perform autonomic functions can be distributed and gathered efficiently highly distributed systems whose end-to-end message route spans many heterogeneous elements. In this scenario the KBN enables distributed elements to publish messages and for autonomic functions to subscribe to them. The approach taken was to extend the routing and subscription matching algorithms of the open source, Siena CBN [12] to use ontology-based subscriptions, hence using ontologically reasoned knowledge for matching and filtering messages inside the network [20].

In Siena, a notification is a set of simply typed attributes. A subscription filter is constructed from a set of filtering constraints which are each applied to these contents of each notification. Where multiple constraints exist in a single filter they are evaluated as a conjunction. An event or notification is delivered to an interested party if it has submitted a subscription filter that matches the notification. A filter **f** "covers" another filter **f'** where together the set of constraints in **f** are more general than all of the individual constraints in **f'**, and so all of the notifications that would be covered by **f'** would also be covered by **f**, i.e. **f** is more general than **f'**.

By exploiting these covering relationships in Siena, efficiency in routing can be gained through subscription aggregation and merging, where routes to subscribing clients are multiplexed with ones with covering subscriptions, i.e. broader subscriptions that will match all the event messages that would match the covered subscription. As such, subscriptions covered by previously forwarded subscriptions are pruned and network traffic is kept to a minimum.

The main change to the subscription language was the addition of three new ontological operators for ontological subsumption and equivalence [20]. The subsumption relationship describes how an ontological entity is more general than another ontological entity. For example, as seen in the Wine ontology [21], the ontological type "wine" subsumes the type "white wine", or "white wine" is subsumed by "wine" since "wine" is less specific than "white wine". These operators are: EQU (equivalent to); MORESPEC (more specific than, or is subsumed by); and LESSSPEC (less specific than, or subsumes). Equivalence refers to the relationship between two ontological types that refer to the same type of entity yet may be different ontological classes.

If an event consumer was interested in receiving events about some ontological entity ***E***, equivalent classes, 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***).

While this may seem to make the subscription specification more difficult for simple subscriptions, the advantages become apparent for more extensive queries and ontologies. The standard subscription language for Siena, and most content-based networking systems, allow filters to be defined using only basic data types, and then only as a conjunction of filters. Filter constraints are combined using the boolean AND operator and so the failure of one constraint in a filter results a match failure for that filter. To specify subscriptions using ontological classes without these new operators would entail the specification of multiple individual subscriptions to match for each class type specified as a string comparison. Additionally, the use of ontological reasoning supports the consideration of equivalent and disjoint classes within the class hierarchy.

The main consideration behind enabling ontology based subscriptions in such a manner is the preservation of the covering relation between filters. In particular, the efficiencies gained from these covering relationships must be maintained. In order to accomplish this we must define a covering relation between our enhanced subscriptions.

Consider two ontological filtering constraints ***A*** and ***B***, such that ***A*** is given as (**x** *op* **a**), and ***B*** is given by (**x** *op* **b**), where *op* is one of EQU (equivalent to), MORESPEC (more specific than, or is subsumed by), or LESSSPEC (less specific than, or subsumes). The variable **x** is the variable for the field in each notification to be compared to the constant ontology class names **a** or **b**, given in the filter specification. Table 1 describes when filter constraint **A** covers filter constraint **B,** i.e., when the set of possible notifications matching filter constraint **A** is a superset of the set of notifications matching filter constraint **B**.

Comment [JK1]: *TABLE I: Covering relationships between new Siena ontological operators*

It should be noted that (**x** MORESPEC **y**) is equivalent to (**y** LESSSPEC **x**). In this design a constraint does not cover itself or an equivalent constraint. The covering relationships for the other Siena operators are given in [12][22], and remain completely unaffected by the addition of the three new operators described here. These covering relationships of filters (containing both normal and ontological constraints) maintain the efficiencies gained by subscription aggregation within Siena.

4. EVALUATION AND DISCUSSION

In order to demonstrate the effects of adding support for ontological operators to the Siena subscription language a number of factors were evaluated. These include: the time taken to load, parse, and reason over a number of ontologies; the effect on scalability and end-to-end time of incorporating ontological lookups in the notification forwarding algorithm; and comparing a sample ontological subscription to an equivalent subscription which only operates on class names using basic string comparison operations.

We envisage that our initial distributed autonomic system will rely on the same management information bases that network and enterprise management system use currently. Therefore, we have used the Distributed Management Task Force's (DMTF) Common Information Model CIM [24] as a standardized example of management knowledge. In CIM, when a management event occurs, its occurrence is signaled to a registered set of interested parties by the creation and dispatch of "indication" objects defined in the CIM Managed Object Format (MOF).

For these experiments, a new indication class (JK_SampleEvent) that is inherited from the standard indication type CIM_AlertIndication, an indication type used to describe error or alert type events. Tools and research from the Universidad Politécnica de Madrid demonstrates the value of modeling management information models in the OWL ontological format to support interoperability between models in different management information languages [23]. By making use of a CIM to OWL conversion utility developed by UPM [23] an OWL ontology

(JK_SampleEvent ontology) was generated that includes CIM Core Model and the CIM Event Model.

For the purposes of evaluating the overhead involved in the load time parsing and inference of an ontology, three ontologies were compared with three levels of reasoning. Firstly the very simple OWL-S Service ontology [25], with only four classes and no individuals; next the complex Wine ontology [21], with 138 classes, and 206 individuals; and finally the large but relatively simple JK_SampleEvent ontology based on the CIM ontology discussed above, with 147 classes and 563 individuals. These ontologies were loaded, parsed, and reasoned over using the Jena framework [26] with three different levels of reasoning. The first reasoner, "OWL_MEM_NONE", supplied with Jena, performed no reasoning. The second "OWL_MEM_RDFS_INF", also supplied with Jena, performs RDFS entailment reasoning. The third reasoner, Pellet [27] performs full OWL DL reasoning. The results of these comparisons are given in Table II.

As can be seen from these results, and further experiments [10], the operations to load an ontology, and especially reason over its contents, are expensive operations. However, in the case where the set of ontologies to be used are known a priori, the loading and reasoning can be performed at initialization time rather than during the operation of the system [10]. When the set of ontologies used changes during runtime, and so need to be reloaded, such changes must be minimized to maintain satisfactory performance. In addition to the size of an ontology, the time to reason over an ontology is dependent on the level of reasoning required for correct interpretation of the ontology, which can be dependent on the complexity of the particular ontology. For a complex ontology, such as the Wine ontology, a more functional reasoner like Pellet is required to obtain a correct class hierarchy, however, for a relatively simple ontology, the full support of Pellet is not required to obtain a correct class hierarchy and can be provided by the less functional OWL_MEM_RDFS_INF reasoner, with a substantial time saving. For this reason it is necessary to carefully tune the specific level of reasoning required to each specific ontology on an application by application and a case by case basis [28]. Further information on the comparative performance of a number of reasoners is available from [29] and [10].

Comment [JK2]: *TABLE II: Time in ms. to load, parse and reason over three ontologies (std. dev. in parenthesis)*

To further evaluate the impact of adding ontological operations to the subscription matching and notification forwarding algorithms in each Siena node, it was necessary to determine how such operations affect the scalability of the Siena network and end-to-end time taken for notifications to be delivered.

For this experiment an open source CIM object manager (CIMOM) [30] was extended to additionally publish a standard Siena notification each time an event occurred. This notification message included the ontological class name of the particular CIM event indication instance created to signify the occurrence of the event. This Siena notification was then published to a testbed network of extended Siena nodes. The ontology used was the extended JK_SampleEvent CIM ontology. A notification subscriber with a specific interest in CIM event indications was then connected at varying locations within the Siena network in a manner to force the Siena notification to traverse a specific number of Siena router hops.

Firstly a single subscription filter, with one constraint, was created to subscribe to all CIM event indications where the ontological class name of the event was more specific (MORESPEC) than CIM_Indication, since all CIM event indications are more specific than this class. This subscription requires that at every Siena node, for every message describing a CIM event indication, the JK_SampleEvent ontology is queried by the routing algorithm. The end-to-end time for notification delivery and the scalability consequences of these operations are presented in Figure 1. As expected the end-to-end delivery time scales linearly.

Secondly, in order to duplicate the same experiment without the use of ontological operators, a number of equivalent string based subscriptions were formed. Since 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) can only be specified using multiple distinct subscriptions. This requires that in order to subscribe to notifications containing names of any of the 16 subclasses of CIM_Indication, 17 distinct subscriptions are required, each causing subscription churn within the Siena network. The end-to-end delivery times of the same CIM event notifications, to a similar client, subscribing using these 17 subscription filters instead of the single ontological subscription, are also given in Figure 1 as a comparison to the use of equivalent ontological operations in the subscription and forwarding algorithm, and are shown to be similar.

Furthermore, when not using ontological operators, if a new subclass of CIM_Indication is added at runtime, instead of simply modifying the ontology at each Siena router node, the subscription code for each subscriber application would need to be changed to explicitly subscribe to that new class, rather than allowing it to be automatically categorized for the ontological operators.

Comment [JK3]: *FIGURE I: KDS Scalability and notification end-to-end timing*

5. CONCLUSION

This paper gives an initial evaluation of the performance impact of introducing ontological operators into an existing decentralized CBN. The performance decrease observed can be seen as acceptable when offset against the increased flexibility of the system, particularly over a short number of hops. Our future work will involve a developing a control plane mechanism for clustering KBN nodes by semantic closeness in order to minimize the number of ontologies needed at any one node and to maximize the likelihood of covering relationships to minimize the forwarding table size as outlined in [10], which we will evaluate against the KBN benchmarking systems we are developing [19] [31]. By supporting arbitrary semantics in the structuring of messages and the construction of consumer subscriptions, any such KBN would provide a stable basis for the long term evolution of new autonomic solutions for highly distributed systems.

6. ACKNOWLEDGEMENTS

This work was partially funded by the EU under the ACCA project, by Science Foundation Ireland under the CTVR project, and by the Irish Higher Education Authority under the M-Zones program.

7. REFERENCES

1. Kephart, J., Chess D., "The vision of autonomic computing," IEEE Computer, vol. 36, no. 1, pp. 41–50, 2003.
2. Mullany, F., Ho, L., Samuel, L., Claussen, H., "Self-Deployment, Self Configuration: Critical Future Paradigms for Wireless Access Networks", International Workshop on Autonomic Communications, Berlin, Germany, Oct 2004
3. Smirnov, M.: "Autonomic Communication: Research Agenda for a New Communication Paradigm", in Fraunhofer FOKUS White Paper, November 2004, http://www.autonomic-communication.org/publications/doc/WP_v02.pdf
4. Clark, D., Partridge. C., Ramming, J.C., Wroclawski, J.T. "A Knowledge Plane for the Internet", SIGCOMM'03, 25-29 August 2003, Karlsruhe, Germany.
5. OWL Web Ontology Language Reference, available at: http://www.w3.org/TR/owl-ref/.
6. Berners-Lee, T., Hendler, J., Lassila, O., "The Semantic Web", Scientific American, May 2001
7. Mulvenna, M., Zambonelli, F., "Knowledge Networks: the nervous system of an autonomic communication infrastructure", International Workshop on Autonomic Communications, Athens Greece, Oct 2005

8. Stevenson, G., Nixon, P., Dobson, S, “Towards reliable wide-area infrastructure for context-based self-management of communications”, International Workshop on Autonomic Communications, Athens Greece, Oct 2005
9. Lewis, D. O’Sullivan, D. Power, R. Keeney, J. “Semantic Interoperability for an Autonomic Knowledge Delivery Service”, International Workshop on Autonomic Communications, Athens Greece, Oct 2005
10. Lewis, D., Keeney, J., O'Sullivan, D., Guo, S., "Towards a Managed Extensible Control Plane for Knowledge-Based Networking", to appear, International Workshop on Distributed Systems: Operations and Management Large Scale Management, (DSOM 2006), Dublin, Ireland, 23-25 October 2006
11. Meier, R., Cahill, V., “Taxonomy of Distributed Event-Based Programming Systems“, The Computer Journal, vol 48, no 5, pp 602-626, 2005
12. Carzaniga, A., Rosenblum, D. S., and Wolf, A. L., “The Design and Evaluation of a Wide-Area Event Notification Service”, ACM Transactions on Computer Systems, vol 19, no 3, August 2001
13. Segall, B., Arnold, D., Boot, J., Henderson, M., Phelps, T., “Content-Based Routing in Elvin4”, In Proceedings AUUG2K, Canberra 2000.
14. Pietzuch, P., Bacon, J., "Peer-to-Peer Overlay Broker Networks in an Event-Based Middleware". International Workshop on Distributed Event-Based Systems (DEBS'03). San Diego, California, June 2003
15. Chand, R., Felber, P.A., “A Scalable Protocol for Content-Based Routing in Overlay Networks”, Second IEEE International Symposium on Network Computing and Applications , Cambridge, MA, April 2003
16. Strom, R., Banavar, G., Chandra, T., Kaplan, M., Miller, K., Mukherjee, B., Sturman, D., Ward M., “Gryphon: An Information Flow Based Approach to Message Brokering“, International Symposium on Software Reliability Engineering, 1998
17. Carzaniga, A., Rosenblum, D., Wolf, A.L., “Challenges for Distributed Event Services: Scalability vs. Expressiveness” Engineering Distributed Objects (EDO '99), ICSE 99 Workshop, Los Angeles CA. May 1999
18. Petrovic, M., Burceaa, I., Jacobsen, H.A. “S-ToPSS – a semantic publish/subscribe system”, Very Large Databases, (VLDB’03), Berlin, Germany, September 2003

19. Keeney, J., Lewis, D., O'Sullivan, D., Roelens, A., Boran, A., Richardson, R., "Runtime Semantic Interoperability for Gathering Ontology-based Network Context", IEEE/IFIP Network Operations and Management Symposium (NOMS 2006), Vancouver, Canada, April 2006
20. Lynch, D., Keeney, J., Lewis, D., O'Sullivan, D., "A Proactive approach to Semantically Oriented Service Discovery", Workshop on Innovations in Web Infrastructure (IWI 2006), Edinburgh, Scotland. May 2006.
21. W3C: The Wine Ontology, http://www.w3.org/TR/owl-guide/wine.rdf
22. Covering relationships in Siena, http://serl.cs.colorado.edu/~carzanig/Siena/forwarding/ssimp/namespaceSiena.html#a1
23. López de Vergara, J.E., Villagrá, V.A., Berrocal, J, "Semantic Management: advantages of using an ontology-based management information meta-model", Proceedings of the HP Openview University Association Ninth Plenary Workshop (HP-OVUA'2002), distributed videoconference, 11-13 June 2002
24. Common Information Model v 2.10.1, DMTF 2005: http://www.dmtf.org/standards/cim/cim_schema_v2101
25. Service Ontology, Part of "OWL-S: Semantic Markup for Web Services", The DAML Service Coalition, http://www.daml.org/services/, October 2002, http://www.daml.org/services/owl-s/1.1/Service.owl
26. Carroll, J., Dickinson, I., Dollin, C., "Jena: Implementing the Semantic Web Recommendations", World Wide Web Conference 2004, 17-22, New York, NY, May 2004. http://jena.sourceforge.net/.
27. MINDSWAP. Pellet: An OWL Reasoner. http://www.mindswap.org/pellet
28. Keeney, J., Lewis, D., O'Sullivan, D., "Benchmarking Knowledge-based Context Delivery System", International Conference on Autonomic and Autonomous Systems (ICAS'06), Silicon Valley, USA, July 2006
29. "Pellet Performance", http://www.mindswap.org/2003/pellet/performance.shtml
30. WBEM Services, http://wbemservices.sourceforge.net
31. Lewis, D., O'Sullivan, D., Keeney, J., "Towards the Knowledge-Driven Benchmarking of Autonomic Communications", International IEEE WoWMoM Workshop on Autonomic Communications and Computing (ACC 2006), Niagara-Falls / Buffalo, NY, June 2006.

SHORT BIOS

John Keeney holds a BAI degree in Computer Engineering and a PhD in Computer Science from Trinity College Dublin. His primary interests focus on controlling autonomic adaptable systems, particularly when those systems are distributed.

David Lewis graduated in Electronics Engineering from the University of Southampton and gained his PhD in Computer Science from University College London. His areas of interest include integrated network and service management, distributed system engineering, adaptive and autonomic systems, semantic services and pervasive computing.

Declan O'Sullivan was awarded his primary degree, MSc and PhD in Computer Science from Trinity College Dublin. He has a particular interest in the issues of semantic interoperability and heterogeneous information querying within a range of areas, primarily network and service management, autonomic management, and pervasive computing.

TABLE 1, MIDDLE OF PAGE 9

	A covers	B	Condition
1	**x** EQU **a**	**x** EQU **b**	***never***
2	**x** MORESPEC **a**	**x** EQU **b**	*if (* **a** LESSSPEC **b** *)*
3	**x** LESSSPEC **a**	**x** EQU **b**	*if (* **a** MORESPEC **b** *)*
4	**x** EQU **a**	**x** MORESPEC **b**	***never***
5	**x** MORESPEC **a**	**x** MORESPEC **b**	*if (* **a** LESSSPEC **b** *)*
6	**x** LESSSPEC **a**	**x** MORESPEC **b**	***never***
7	**x** EQU **a**	**x** LESSSPEC **b**	***never***
8	**x** MORESPEC **a**	**x** LESSSPEC **b**	***never***
9	**x** LESSSPEC **a**	**x** LESSSPEC **b**	*if (* **a** MORESPEC ***b*** *)*

TABLE I: Covering relationships between new Siena ontological operators

TABLE 2, BOTTOM OF PAGE 12

Reasoner	Loading and Reasoning			Loading only		
	Service	***Wine***	***JK_SampleEvent***	***Service***	***Wine***	***JK_SampleEvent***
OWL_MEM _NONE	52.63 ms (*2.27*)	329.64 ms (*9.95*)	455.38 ms (*24.88*)	57.60ms *(9.21)*	338.84 ms *(15.41)*	430.67ms *(11.78)*
OWL_MEM _RDFS_INF	61.32 ms (*2.67*)	366.56 ms (*6.93*)	579.35 ms (*29.40*)	55.37 ms *(4.38)*	346.3 ms *(10.91)*	444.68 ms *(15.00)*
PELLET	97.98 ms (*9.69*)	1391.39 ms (*29.30*)	1064.54 ms (*48.18*)	58.91 ms *(11.46)*	364.48 ms *(18.20)*	441.5 ms *(18.50)*

TABLE II: Time in ms. to load, parse and reason over three ontologies (std. dev. in parenthesis)

FIGURE 1, BOTTOM PAGE 13

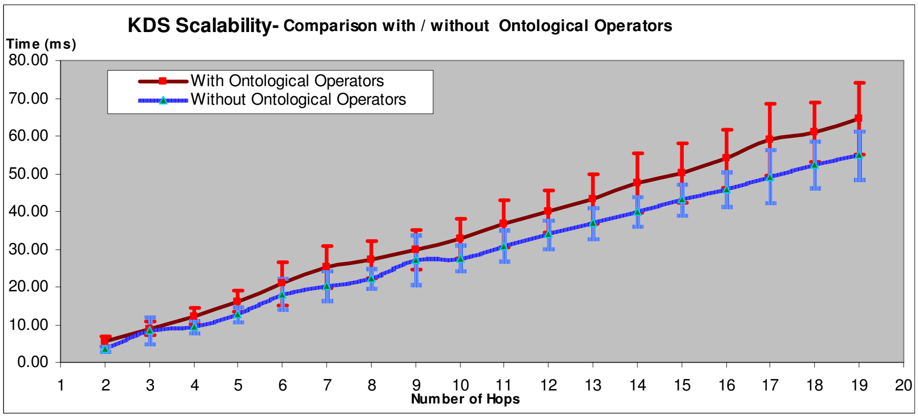

FIGURE I: KDS Scalability and notification end-to-end timing

TABLES

TABLE I: Covering relationships between new Siena ontological operators

TABLE II: Time in ms. to load, parse and reason over three ontologies (std. dev. in parenthesis)

FIGURES

FIGURE I: KDS Scalability and notification end-to-end timing