

Semantic Interoperability for an Autonomic Knowledge Delivery Service

David Lewis, Declan O'Sullivan, Ruaidhri Power, John Keeney

Knowledge and Data Engineering Group
Trinity College Dublin, Ireland

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

Abstract: The development and deployment of interconnected networks is being increasingly limited by their complexity and the concomitant cost of managing the operational network. Autonomic Communication aims to reduce this cost, by migrating management intelligence towards the network elements and empowering operational support staff to specify what network behaviour in terms of goals and constraints. Towards this aim we propose a key infrastructural service that enables the efficient delivery of network operations knowledge to, and only to, nodes that have expressed an interest in that knowledge. This Knowledge Delivery Service mediates operational network knowledge in an open, ontological form, thereby promoting the graceful evolution of network management applications from contemporary to fully autonomic. To cope with the inevitable heterogeneity of knowledge across the population of network nodes, the service provides a level of semantic interoperability that will be transparent to the nodes providing and consuming knowledge. The Service will be based on content-based networking principles. This paper describes work towards supporting semantic interoperability in such a Knowledge Delivery Service.

1. Introduction

The development and deployment of interconnected networks is being increasingly limited by their complexity and the concomitant cost of managing the operational network. Autonomic Communications aims to reduce this cost, by migrating management intelligence towards the network element and empowering users and operational support staff to specify network behaviour in terms of goals and constraints, rather than specifying how that behaviour should be achieved.

Autonomic principles are targeted at reducing the cost of handling the complexity of distributed computing systems by making them self-managing, i.e. self-configuring, self-healing, self-optimising and self-protecting [kephart]. This requires monitoring and analysing the operational knowledge in systems so that it can be used to plan and execute corrective measures, typically using some artificial intelligence techniques. This relieves the human manager from performing these tasks while allowing the human to guide the decisions made by the autonomic manager through

the definition of high-level policy rules defining goals for and constraints on the desired system behaviour.

More recently, momentum has been growing to apply autonomic principles to network operations, i.e. Autonomic Communications. An early articulation of the use of operational network knowledge by intelligent applications was proposed by David Clarke et. al. in a proposal for a Knowledge Plane for the Internet [clark]. Operational network knowledge is defined as network operations or management data accompanied by its meta-data, typically expressed as a management information model. These approaches present a major challenge in obtaining the relevant operational network knowledge. Difficulties arise because the network elements that possess this knowledge are widely distributed, they are purchased from different vendors, they perform different functions, they possess a wide range of knowledge meta-data and, perhaps most challenging of all, they are operated by different organisations.

Current approaches to Autonomic Communication typically involve distributed intelligence, such as multi-agent systems, swarm intelligence, or cellular automata [mullany], operating at the network element level, adapting to changes in the knowledge that is gathered on the network and application context. This adaptation is constrained by policies representing the operational goals and constraints of network operators and users. 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. This work tackles head-on the interoperability short-comings of current Autonomic Communication proposals, but in a way that ensures a smooth, commercially viable transition from contemporary network management systems to fully autonomic ones. We therefore propose a Knowledge Delivery Service (KDS) as an infrastructure that accurately and efficiently delivers autonomic network knowledge to nodes that have expressed an interest in that knowledge. Here we focus on how this service might adapt the knowledge delivered between the semantics used by the producer and those expected by the consumer. This paper describes the major technical challenges in developing semantic interoperability in a KDS and then presents initial results on how semantic interoperability knowledge can be captured and then distributed and used within such a service.

2. Background

The proposed Knowledge Delivery Service presents a demanding set of challenges that intersect Semantic Web, Content-based Networking and attribute based access control research.

Communication service operator concerns about the sensitivity and security of operational data is reflected in the hierarchical nature of the manager-agent paradigm and the intra-domain focus of architectures such as TMN (Telecommunications Management Network) [TMN]. The fragmentation of manager-agent protocols at the element layer, and the lack of a dominant interoperation technology at the higher

layers has led to problems exchanging management knowledge between the vertical silos of interoperability (both syntactic and semantic) within operators' Operational Support Systems (OSS) [adams]. When exchanging operational knowledge between operators, this is compounded by commercial confidentiality concerns, which result in bilateral agreements and inflexible custom gateways. This spells disaster for the vision of Autonomic Communications where intelligent agents operating at or near the network element level must be able to freely gather contextual knowledge about the state of the network end-to-end and adapt to changes in this context to achieve administrator-specified goals and maintain their constraints. The challenge of cross-ownership sensitivities is addressed elsewhere [feeney05], while here we focus on the interoperability issues in gather network context to guide the behaviour of autonomic network elements. These present an even more extreme case of the conditions that led to interoperability silos in conventional OSS and thus the looming prospect of Autonomic Communications silos must be urgently addressed. Proposals for end-to-end delivery of operation network knowledge are either constrained to individual protocol layers [thaler] or to following existing signalling paths [schulzrinne]. These approaches are, however, insufficient as the wider network state increasingly forms the context for intelligent decision making in network elements [karmouch]. Any scalable solution will rely on loose coupling between the producers and consumers of autonomic knowledge. In this work we focus on late semantic binding through the encoding and mapping between heterogeneous models using the existing Web Ontology Language (OWL) [owl]. As a W3C standard OWL represents a broadly applicable mean for capturing semantics with basic language primitive for capturing semantic mapping, as well as providing the basis for richer mapping languages. This is coupled with loose binding between producers and consumers of operational network knowledge using publish-subscribe communications.

Several attempts to address management model heterogeneity have been made by defining new management information modelling languages to act as a canonical model providing lingua franca between other models. Notable amongst these are the Distributed Management Task Force's (DMTF) Common Information Model (CIM) schema [cim] and the TeleManagement Forum's NGOSS technology neutral architecture [tmf053]. However, a lack of a strong semantic interoperability mechanism and reliance on conformance to poorly subscribed industrial agreements effectively render these as yet more management knowledge formats with which other schemes needed to interoperate. Recent pioneering work by Vergara and Villagra [lopezdevergara] has shown directly the value of modelling management information models in the OWL ontological format, and how this can be used to ease the interoperation between models originally conceived in different management information languages, i.e., GDMO, SMI, CIM. Our approach follows the underlying philosophy of the semantic web, where semantics, including mappings, are captured where applications require it and the necessary expertise is present. Our aim, however, is to ensure what mappings exist are made available as automatically as possible to the management applications that can use them. This also points to the adoption of the OWL-S service semantic language [martin] for defining management services knowledge as well as for supporting to dynamic management service composition.

Publish-subscribe systems provide an efficient mechanism for delivering information from its source to one or more interested parties (known as subscribers). It also allows the timely notification of events or changes to information, when compared to polling approaches, but requires publishers and subscribers to agree on the message types before interacting. Content-Based Networks (CBN) extend this approach to allow the subscriber to specify conditional filters on message properties, effectively allowing the subscriber to define the type of message in which they are interested [carzaniga][segall][strom]. By delivering operational knowledge only to those knowledge consumers who register a specific interest, while multicasting messages to consumers who share interests offers the potential for scaling knowledge delivery to Internet dimensions [crowcroft]. We propose a KDS that will be implemented as a Knowledge Delivery Network (KDN) structured along CBN principles. Thus, network elements may advertise the type of knowledge they possess to the KDS while an intelligent autonomic network element may place a subscription for the knowledge they need for the task at hand, cancelling the subscription when the task is finished. Producers of operational network knowledge express this capability using the ontological representation of the relevant management information models and while consumers express subscriptions as simple semantic queries. The advertisement-subscription mapping and subscription aggregation algorithms used in the KDN may therefore exploit ontology-based reasoning mechanisms, such as class subsumption. In this paper we will focus on how ontological mappings between management information models will be used by the KDN nodes to supplement these algorithms providing a level of semantic interoperability between the models of the autonomic context knowledge sought by the consumer and that used by its producer.

In the rest of this paper we focus on how mappings between managed object concepts on heterogeneous systems can be captured and then injected into the KDS to support the dynamic semantic interoperability of management knowledge.

3. Semantic Interoperability for Knowledge Delivery

Semantic Interoperability is a key element of the KDS. Where the semantics of information emitted by a notification producer does not immediately conform to the semantics sought by a particular consumer's subscription, a match may still be possible if the knowledge exists of how one set of semantics may be mapped to the other. Mapping one knowledge domain onto another typically requires human comprehension of both sets of semantics, though tools are increasingly able to produce mappings by extrapolating from a few human supplied mapping anchors. More automated collaborative identification of semantic mapping is being researched using intelligent agent technologies.

In the context of the KDS, the service must operate in a framework that supports the discovery, injection and interpretation of known mappings, regardless of whether they are human-designed or auto-generated. Mapping interpretation is then undertaken at runtime to route information appropriately and to aid the transformation of information between two different formats.

3.1 Discovering Semantic Mappings

A key challenge is how the mapping information between the ontologies can be derived. Automatically deriving ontology mapping information at runtime without the involvement of a human is generally considered impossible [klein] due to diversity of domains and lack of encoded semantics. The network management domain has better semantic homogeneity than many due to its standardised information models that represent the common resource semantics needed for control and user plane interoperability. However, differing management standards and proprietary extensions, which representing competitive differentiations inherent in the industry, combined with the frequent need to integrate arbitrary elements within increasingly complex network still present difficult semantic interoperability challenges. The challenge in our work, therefore, has been to identify an integrated software and process framework which will minimise the amount of design time work involved, devolve as much work as possible to a runtime algorithm, and share the mappings as much as possible such that human involvement is reduced. This devolution is crucial for the uptake of this approach in autonomic computing environments where reduction of human intervention is key. Equally important is maximising the applicability of human generated ontology mappings by ensuring that they maximise the chances of a successful runtime mapping between information conforming to concepts from the two ontologies concerned.

The process of the resultant OISIN (Ontology Interoperability for Semantic Interoperability) framework [vanderMeer] is overviewed in Figure 1 and illuminated further with an example. This example involves interoperability between intelligent agents that are resident on the management agents of different printer types and are aware of the local printer information models. A useful application here may entail agents monitoring neighbouring printers in an office for out-of-paper notifications, and conserving its paper (e.g. by automatically switching to double sided or two up) to maintain office-wide print services until notification of fresh supplies being loaded have been detected. In this example we assume printer agents have knowledge of either a local SNMP printer MIB or the DMTF CIM printer device model.

In the first phase of the OISIN process the ontologies from each party are characterised. These ontologies represent the core concepts that would be used in self-management functions in the two agents types, that is a CIM based ontology and a Printer MIB based ontology. Of course it is assumed in future that these ontologies would be pre-existing, perhaps through the use of a conversion tool such as presented in [Lópezdevergara].

The tools of the Characterisation Phase transform the ontology (in Ontology Web Language (OWL) format or relational database format, etc.) into a common internal format. The software tools in this phase characterise:

- The nature of the terms, whether simple or composite. This can be helpful in determining whether finding mappings will be straightforward for the mapping algorithm and the complexity of the human driven confirmation process of the algorithm suggested mappings;
- How many terms are known/unknown by WORDNET the online dictionary developed in Princeton. This can be helpful in evaluating the degree of domain or acronym specific terminology;

- The quality of the ontologies according to the online Semantic Web search engine SWOOGLE developed at the University of Maryland. The information from SWOOGLE can provide an indication as to how widely referenced a particular ontology is, which provides one measure of quality;
- The number of candidate matches. Class and property names of the ontologies are compared (with support of WordNet and an encoded telecommunications domain specific thesaurus) to identify potential matches (through exact or synonym matches) of ontology classes and their properties. In the initial implementation a lexical matcher is used, but this is being extended to type and range matchers
- The number of potential mappings arising from the candidate lexical matches. This information can provide an indication of the amount of overlap between the ontologies, and also provides information about the potential difficulty in finding/confirming mappings from suggested matches.

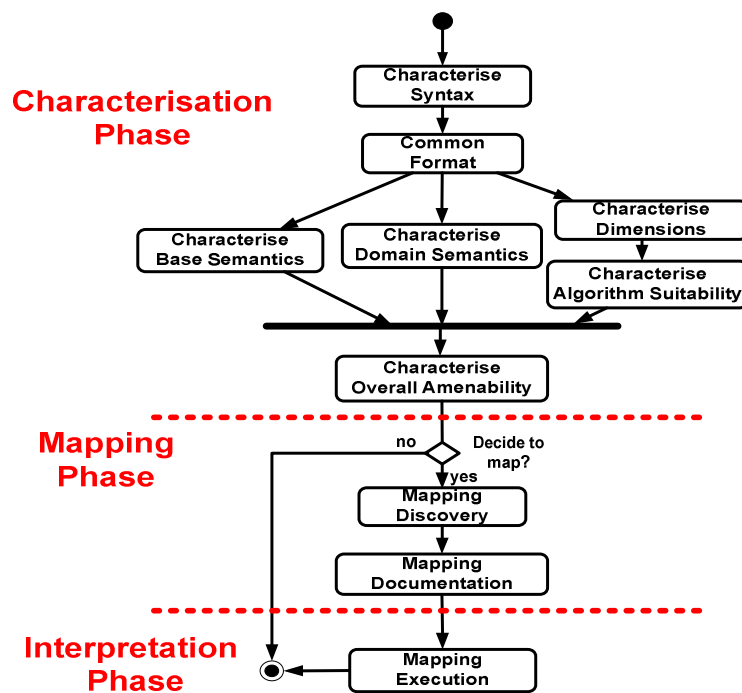


Figure 1: OISIN Process Overview

The characterisation information generated by the **Characterisation Phase** is presented to the user via numeric and graphical charts to the user, so that a decision to map or not can be made. If it is decided to map then in the Mapping Phase the matches are presented to the user in a graphical manner. In Figure 2 for example the M identifies exact lexical matches (e.g. Printer) and the P identify partial matches on

a lexical or synonym basis (e.g. *Person* partially matches on a partial synonym basis to *Operator*, *Manager* and *User*).

The user then identifies the “anchors” which correspond to key partial mappings between the ontologies. This involves examining the two ontologies to try to identify equivalent concepts. Typically during this examination the properties of the concepts are examined to identify equivalence as well. In the example shown in Figure 2 the *MaxNumberUp* property of the CIM Printer class can be seen to be equivalent to the *prtOutputMaxCapacity* property concept of the MIB Printer class. Once an anchor is chosen it is annotated with an E (e.g. *Printer* in Figure 2). In addition, XSL based transformation code/bridge can be associated with a mapping in order to provide the ability to translate from one value range to another.

During this process “anchor paths” are also identified. The concept of “anchor paths” was first introduced by [noy01]. The idea is that if two anchors are specified in a hierarchy of Ontology *A* it is likely that the classes which appear in the intervening path may correspond with those on the path of the corresponding anchors in Ontology *B*. Unlike the PROMPT/Anchor tool [noy00], we do not actually require the user to enumerate every mapping in these paths at design time but devolve the determination of what is or is not a match within these paths to the runtime algorithm.

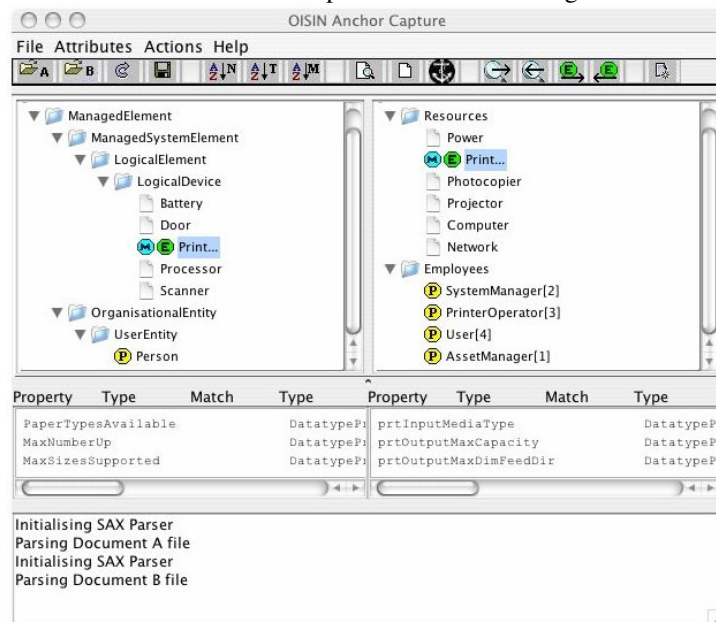


Figure 2: Example of OISIN capture tool

A key differentiation of our approach in general is our belief that the determination of what is or is not considered class equivalence can only be undertaken in the context of the applications involved in using the mappings and what they are trying to achieve. For this reason the original matching information as well as the equivalence annotations provided by the user are made available to an application that uses the mapping information. The output of this phase is mapping information consisting of a set of anchor mappings (expressed using the `owl:equivalentClass` and

owl:equivalentProperty XML elements) and their corresponding XSL based transformation bridges. In [osullivan] we have shown how XSL based transformation bridges can be automatically generated given a set of ontology mappings, and so we will not provide further detail here on the bridge creation. In addition the mapping information output contains the matches information and the anchor path information that has been generated, which are used at runtime by a Semantic Matching Utility when no default transformation bridge exists. The Semantic Matching Utility returns the anchor path and matching information for a term requested and this can be used by an application to dynamically create a transformation depending on the context. In the **Mapping Interpretation** phase, the mapping information is injected into the KDS for appropriate mapping execution. The transformation bridges and the Semantic Matching Utility are used by the Knowledge Discovery Service Node (KDSN) during query resolution (see next section). As discovery of such mappings is likely to be a decentralised task, the distribution of mappings to points in the KDN where they are needed for interpretation purposes is itself performed using the content-based networking feature of the KDN.

3.3 Mapping Interpretation

In our architecture, the KDS is provided to intelligent network elements to make autonomic knowledge available to them. The role of a KDS is to take queries from a client and to resolve those queries by acting as a mediator between the client and other knowledge sources that the service has access to. As well as acting as consumers of autonomic knowledge (by executing queries), intelligent NEs can also act as producers of autonomic knowledge. In our current implementation interaction between KDS client agent applications and the KDN is implemented as an API. The implementation of the API operates in the same memory space as the application, but can be considered as a KDS Edge Node (KDSEN). The KDS is not currently designed to define a specific query language and may support multiple query language styles, such as CMIS scope/filer requests, CIM queries, SQL, XQuery, RDQL or SPARQL. Our current implementation uses XPath. The terms involved in the query must be a subset of those in the ontologies understood by the KDS client, for the service to be able to formulate the query and understand the response. Associated with each KDSEN there is a repository of ontologies that describes the domain of knowledge that this agent application currently understands, i.e. the OWL version of the NE's local MIB and other models the agent uses in a manager role. In more intelligent agents this domain may extend during operation as new concepts are used in downloaded policies or as the agent learns concepts from peer agents. These ontologies are provided to the KDSEN by the application agents, either when registering notifications it is able to provide or making query subscription. The KDS aims to resolve query subscription immediately and then update the subscribing application with any matching updates until the subscription is cancelled.

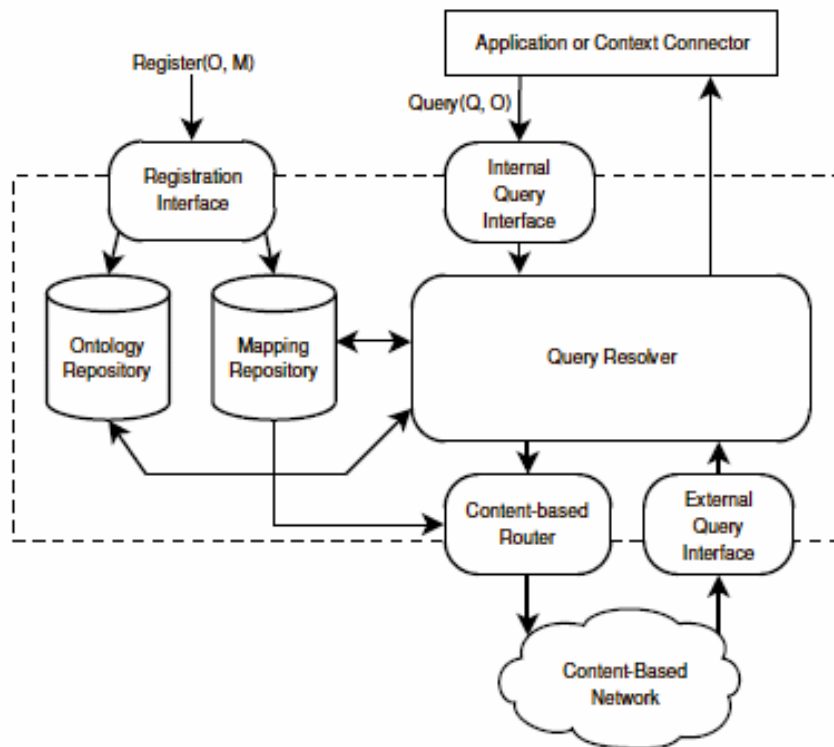


Figure 3: Knowledge Delivery Service Edge Node Architecture

The internal architecture of a KDSEN is shown in Figure 3. The registration interface allows intelligent NE agents to register with a Knowledge Discovery Service Node. During the registration the application will provide a reference to an ontology (O) that defines the domain information in which the agent will couch its queries. The registration interface is also used to inject mappings and bridges from the Mapping Phase of the OISIN process into the KDSEN. The ontology and mapping repositories store the ontologies and bridges that have been registered and the mappings that have been injected at a particular KDSN. The Internal Query Interface takes in queries (Q) using terms from the ontology (O) and passes them to the Query Resolver for resolution. If the application does not have the knowledge required to resolve the query locally, the KDSEN directs it via the internal query interface and a “context connector” is introduced. This essentially is an interceptor which intercepts queries normally destined for the repository that the application normally queries.. The Query Resolver takes this query expressed using terms from the local ontology, passes it to the KDN, and resolves the responses received for that query. The KDN takes the query and routes the query to KDSENs which have directly produced information using the terms involved. In addition, it routes the query to KDSENs where the mapping information injected previously has indicated that relevant information exists. Mappings are themselves distributed by the KDN, with KDSEN’s subscribing to any mapping that include concepts form locally registered ontologies.

The External Query Interface of a KDEN receives a query and undertakes a response. If some transformation has to take place due to the fact that the query arrived as a result routing due to mappings, then the corresponding dynamic bridge to handle the appropriate query and response translations (automatically created during the mapping injection phase) is invoked.

In order to illustrate the above, we return to the printer agent example described earlier. First a semantic mapping is undertaken between the CIM model and SNMP model, one such mapping being the equivalence between the CIM `Printer` and the MIB `Printer` classes. This mapping is then injected into the Knowledge Discovery Service, resulting in nodes which have a MIB based printer attached having a mapping and associated bridge registered. A context connector is introduced so that the application does not need to be altered to take advantage of the KDN. Thus when the application seeks to discover the output capacity of all the printers being managed, the connector poses the query locally via CIM and also passes the query to the KDSN. The KDN then distributes this query to all nodes which either directly supports CIM based printers or those which have been mapped (e.g. via the MIB mapping in our case). Any node which has a MIB based printer attached then receives the query via the external query interface, applies the bridge to transform the query from CIM to SNMP based and transform the responses if necessary into CIM format. The Query Resolver of the KDSN that originally received the query then takes all the responses and returns them.

4. Conclusion and Further Work

This paper presents some initial work in the semi-automated capture of semantic interoperability mappings for use in the run-time translation of Autonomic Communication Knowledge and how those mappings can be integrated into a run-time semantic interoperability system suitable for integration with a Knowledge Delivery Service. We are currently working towards integrating this with a conventional content-based network (Elvin [sutton]). This is being integrated with a commercial element management system, which is being augmented with OWL conversion and translation bridge support that will allow initial performance evaluations. We then aim to extend our investigations into the design of the core KDN. We aim to leverage recent work showing that perfect routing can be achieved in a scaleable manner independently of subscriber joins and leaves through subscription aggregation [chand]. We also aim to address consumption of composite notifications [courtenage].

In any multi-domain scenario those responsible for any knowledge resource must be able to impose access control over who is able to access that knowledge. Access control policies have been demonstrated to work with CBN sources [belokosztolszki]. This employed role-based access control which is the predominant approach to defining access control policies. However, in a fluid, multi-domain scenarios the detailed business modelling that underpins the identification of the roles used will make this approach very brittle. Instead we will adopt a community-based policy approach, which has been shown to track more easily and accurately the dynamic

organisational grouping within and between organisations [feeney04]. This approach will be adapted to support access control of knowledge available to the KDN, including a mechanism to identify possible access control conflicts and suggest resolutions. Such KDN access control requires new mechanisms for matching subscriptions to producer access control rules, without strongly binding subscriptions to consumer attributes and thereby reducing the routing efficiency gained by subscription aggregation. This will be combined with a trust-based access control mechanism for determining community membership in ad hoc organisational situations [feeney05].

Acknowledgements

This work was supported partially by the Irish Higher Education Authority through the M-Zones programme and partly by Science Foundation Ireland and Enterprise Ireland through the Centre for Telecommunication Value Chain Research. The authors would like to thank Simon Courtenage, Thanassis Tiropanis and Victor Villagra for their help in developing these ideas.

References

- [adams] Adams, E., Willetts, K., "The Lean Communication Provider: Surviving the Shakeout through Service Management Excellence", McGraw-Hill, 1996
- [belokosztolszki] Belokosztolszki, A., Eysers, D.M., Pietzuch, P.R., Bacon, J., Moody, K., "Role-based access control for publish/subscribe middleware architectures". In International Workshop on Distributed Event-Based Systems (DEBS03), ACM SIGMOD, San Diego, CA, USA, 2003. ACM
- [carzaniga] Carzaniga, A., Rosenblum, D. S., and Wold, A. L., The Design and Evaluation of a Wide-Area Event Notification Service, ACM Transactions on Computer Systems, Vol. 19, Issue 3, August 2001
- [chand] Chand, R., Felber, P.A., "A Scalable Protocol for Content-Based Routing in Overlay Networks", Second IEEE International Symposium on Network Computing and Applications, April 2003, Cambridge, MA
- [cim] Common Information Model v2.5, DMTF 2000: http://www.dmtf.org/spec/cim_schema_v25.html
- [clark] D. Clark, C. Partridge, J.C. Ramming, J. Wroclawski, "A Knowledge Plane for the Internet", SIGCOMM'03
- [courtenage] Courtenage, S. "Specifying and Detecting Composite Events", In 1st International Workshop on Discrete Event-Based Systems, Vienna, 2002
- [crowcroft] Crowcroft, J., Jean Bacon, J., Pietzuch, P., Coulouris, G., Naguib, H., "Channel Islands in a Reflective Ocean: Large-Scale Event Distribution in Heterogeneous Networks", IEEE Communications Magazine, 40(9):112-115, September 2002
- [feeney04] Feeney, K., Lewis, D., Wade, V., "Policy based Management for Internet Communities", Proceeding of the 5th IEEE International Workshop on Policies for Distributed Systems and Networks, 7-9 June 2004, York Town Hieght, NY, USA

- [feeney05] Feeney, K., Quinn, K., Lewis, D., O'Sullivan, D., Wade, V. "Relationship-Driven Policy Engineering for Autonomic Organizations", to appear in proc. of 6th IEEE International Workshop on Policies for Distributed Systems and Networks, 2005
- [karmouch] Karmouch, et al "Contextware Research Challenges in Ambient Networks", in proc of 1st Intl workshop on Mobility Aware Technologies and Applications (Mata 2004), Florianopolis, Brazil, Oct 2004, Springer LNCS 3284, pp 62-77
- [kephart] J. Kephart, D. Chess "The Vision of Autonomic Computing" Computer, Jan 2003
- [klein] Michel Klein, "Combining and relating ontologies: an analysis of problems and solutions". In Workshop on Ontologies and Information Sharing, IJCAI'01, Seattle, USA, August 4-5, 2001
- [lópezdevergara] López de Vergara, J.E., Villagrà, V.A., Berrocal, J., "Applying the Web Ontology Language to management information definitions", IEEE Communications Magazine, Vol. 42, Issue 7, July 2004, pp. 68-74
- [martin] David Martin, Mark Burstein, Jerry Hobbs, Ora Lassila, Drew McDermott, Sheila McIlraith, Sriniv Narayanan, Massimo Paolucci, Bijan Parsia, Terry Payne, Evren Sirin, Naveen Srinivasan, Katia Sycara, "OWL-S: Semantic Markup for Web Services", W3C Member Submission 22 November 2004
- [mullany] Mullany, F., Ho, L., Samuel, L., Claussen, H., "Self-Deployment, Self Configuration: Critical Future Paradigms for Wireless Access Networks", in Proc of the 1st IFIP International Workshop on Autonomic Communications, Dec 2004
- [noy00] Noy, N.F., Musen, M.A., "PROMPT: Algorithm and Tool for Automated Ontology Merging and Alignment". Seventeenth National Conference on Artificial Intelligence (AAAI-2000), Austin, TX, 2000
- [noy01] Noy, N.F., Musen, M.A., "Anchor-PROMPT: Using Non-Local Context for Semantic Matching". Workshop on Ontologies and Information Sharing at the Seventeenth International Joint Conference on Artificial Intelligence (IJCAI-2001), Seattle, WA, 2001.
- [osullivan] O'Sullivan, D., Lewis, D., "Semantically Driven Service Interoperability for Pervasive Computing", Proceedings of the 3rd ACM International Workshop on Data Engineering for Wireless and Mobile Access, San Diego, CA, USA, 19th September 2003, pp 17-24
- [owl] OWL Guide, W3C, <http://www.w3.org/TR/owl-guide/>
- [schulzrinne] Schulzrinne, H., Hancock, R., "GIMPS: General Internet Messaging Protocol for Signaling", Internet Draft, draft-ietf-nsis-ntlp-05, February 2005
- [segall] Segall, B. et al, "Content-Based Routing in Elvin4", In Proceedings AUUG2K, Canberra 2000
- [strom] Strom et al., "Gryphon: An Information Flow Based Approach to Message Brokering", In International Symposium on Software Reliability Engineering 1998 [smirnov] Smirnov, M., "Autonomic Communications: A Research Agenda for a New Communication Paradigm", November 2004, http://www.autonomic-communication.org/publications/doc/WP_v02.pdf
- [sutton] P. Sutton, R. Arkins, and B. Segall, "Supporting Disconnectedness – Transparent Information Delivery for Mobile and Invisible Computing," in Proceedings of IEEECCGrid 2001. Brisbane, Australia. pp. 277-285. 2001
- [thaler] Thaler, D. G., Ravishankar, C.V., "An Architecture for Interdomain Trouble Shooting", Journal of Network and Systems Management, 12(2), pp 155-189, June 2004
- [tmf053] NGOSS Architecture, Technology Neutral Specification, Membership Evaluation Version 1.51, TeleManagement Forum, July 2001
- [TMN] ITU-T Recommendation M.3010 (1992), Principles for a TMN
- [vanderMeer] van der Meer, S., O'Sullivan, D., Lewis, D. Agoulmine, N., "Ontology Based Policy Mobility for Pervasive Computing", in proc 9th IFIP/IEEE International Symposium on Integrated Network Management (IM 2005), May 2005, Nice, France