Semantically Driven Service Interoperability for Pervasive Computing

Declan O'Sullivan, Dave Lewis

28th May 2003

Abstract

The common vision of pervasive computing environments requires a very large range of devices and software components to interoperate seamlessly. From the assumption that these devices and associated software permeate the fabric of everyday life, a massive increase looms in the number of software developers deploying functionality into pervasive computing environments. This poses a very large interoperability problem for which solutions reliant solely on interoperability standards will not scale. An interoperability problem of a similar scale is presented by the desire for a Semantic Web supporting autonomous machine communication over the WWW. Here, solutions based on service-oriented architectures and ontologies are being actively researched, and we examine how such an approach could be used to address pervasive computing's interoperability problem. The paper outlines the potential role that semantic techniques offer in solving some key challenges, including candidate service discovery, intelligent matching, service adaptation and service composition. In particular the paper addresses the resulting requirement of semantic interoperability outlining initial results in dynamic gateway generation. In addition the paper proposes a roadmap identifying the different scenarios in which semantic techniques will contribute to the engineering and operation of pervasive computing systems.

Keywords

pervasive computing, service composition, semantic interoperability, topic maps, DAML-S.

1 Introduction

Weiser's original vision of pervasive computing* was driven by the move towards increasingly cheap and minaturised computers become embedded into our physical environments (Weiser 1991). The benefits are perceived as ubiquitous access to information, untethered communications based on wireless technologies and computer mediated interaction with environment through sensing, actuating, displaying. This would provide 'intelligent' support for things people want to do through interpreting their needs and task context to support their interaction with software and with the immediate physical world.

This vision, however, presents many technical challenges, which are actively being addressed in the research community;

- Massive scalability in connectivity requirements which is prompting new network and distributed system architectures – e.g. peer-to-peer.
- Heterogeneity of processor forms, such that Windows and Linux address a shrinking 'PC' niche and must compete with small footprint OSs, e.g. PalmOS.

* We use the term 'pervasive computing' synonymously with 'ubiquitous computing' an the term 'pervasive computing environment' synonymously with 'smart space'

- Poor application portability over ‘embedded’ processors, with the larger range of processor types and operating systems increasing software platform dependency and presenting real challenges to Java as a potential solution.
- Heterogeneity of access networks driven by rapid innovation in cellular and wireless LAN and PAN technologies. Multiple wireless standards are therefore emerging, but cheap terminal solutions for simultaneous access may obviate this problem.

However, the source of the most serious challenges to deploying the pervasive computing vision are not technological but structural. Embedding processors, sensors and actuators in everyday products implies an explosion in the number and type of organisations that need to be involved in achieving the seamless interoperability implied by the pervasive computing vision. Many of the network interoperability problems can be addressed by the inter-networking approach of the Internet’s network and transport protocols. However, the potential for debilitating heterogeneity in application level interoperability remains. Consider the complexity involved in reaching agreements on and enforcing conformance to interoperability standards when the players involved expand from the likes of Microsoft, IBM and Cisco to all the potential organisations with applications embedded in their products, e.g. Kellogs, Nike, GAP, Yale, Ford, Pizza Hut, Pentel to name a few implied by pervasive computing scenarios. It is therefore clear that the pervasive computing vision implies a massive increase in scale of the application interoperability problem.

The work presented here is motivated by the observation that we cannot, therefore, rely on shared a priori knowledge via common interoperability standards to solve the application interoperability problems on the scale needed for pervasive computing. Instead, application software must somehow adapt at deployment time and runtime to integrate their functionality and dynamically interoperate with other application software.

The following sections first outline two major evolving technologies that, when combined, show great promise in addressing the interoperability problem involved in engineering pervasive computing systems. The first is service-oriented architectures, which provide a path to dynamically integrating third-party functionality. The second is ontologies, which involve the explicit representation of the semantics of information or functionality in a machine processable form. The paper then goes on to propose how these fields are being integrated in the development of semantic web services. We then outline a roadmap identifying various development-time and run-time challenges in engineering pervasive computing applications and propose that they may be addressed effectively through dynamic semantically-driven interoperability within a service-oriented architecture. Issues related to the engineering of such semantic service are further explored before conclusions are given and future research directions are identified.

2 Service-oriented Interoperability

Service-oriented architectures focus on application level interoperability. This is achieved through the definition of well-defined abstract interfaces to distributed software components to allow their rapid integration into the design of a new application. Service-

---

* We use the term ‘pervasive computing’ synonymously with ‘ubiquitous computing’ an the term ‘pervasive computing environment’ synonymously with ‘smart space’

oriented approaches support explicit bindings between the abstract interface definition and different communication protocols or other distributed interaction technologies. This allows the same abstract service definition to be used in several technology domains, often allowing the same core software functionality to be reused also. The combination of an abstract interface definition and its binding to a concrete interaction technology constitutes a service. Service-oriented architectures often support the concept of reflection. This allows an instance of a service to reveal at runtime its service definition, thus supporting run-time binding between services by any agent able to understand and make decisions based on the service definitions.

One of the main concrete examples of an open service-oriented architecture was the OMG's Object Management Architecture, which included CORBA and a number of general purpose and domain specific application services. The OMA supported reflection though its dynamic invocation interface. It also defined the General Inter-ORB Protocol which could be mapped to different transport protocols, the binding most widely used being the Internet Inter-ORB Protocol. CORBA suffered however, from a use of IP that could not be easily deployed over firewalls, thus preventing its effective deployment over the Internet.

The proven inter-domain capabilities of WWW protocols has led to the application of service-oriented architecture over the Web in the form of Web services. This application of a service-oriented architecture is attracting huge research and commercial interest as it is seen as a major key to seamless e-commerce over the Internet. This has resulted in a number of standardisation efforts. Most notable is the World Wide Web Consortium's (W3C) standardisation of the Web Service Description Language (WSDL). This is a relatively simple XML language for defining services in terms of operation names and their inputs and output. WSDL services, called port types, can be bound to any suitable communication protocol, though a binding to SOAP over HTTP has been standardising and is proving the most popular for ecommerce applications over the WWW.

The standardisation of WSDL, the strong support for generating WSDL-based services on many application server platforms and the emergence of WSDL-services on the WWW offers the possibility of rapidly developing new services trhough combinations of existing ones. The process of organising a set of existing services into a new service is called service composition. Service composition is of interest where services from different organisations may be integrated into some value added service and also for enterprise application integration, where services from different, often distributed, organisational units need to be rapidly integrated to meet changing business needs. The service-oriented view of such integration enables it to be done initially at the abstract interface level and then implemented with a large degree of automation, for example using workflow execution techniques. A wide range of languages for defining such service composition have appeared, e.g. BPEL4WS, WSFL, BPML etc. However, as pointed out in (van der Aalst 2003), these offerings seem primarily bound to existing platforms and associated commercial interests and don't exploit the formal process semantics offered Petri nets and process algebras. Such languages may therefore prove insufficient for the automated service composition.

---

* We use the term 'pervasive computing' synonymously with 'ubiquitous computing' an the term 'pervasive computing environment' synonymously with 'smart space'

2.1 Challenges of Service Composition in Pervasive Computing

The vision of pervasive computing offers many of the same challenges as are already being addressed by dynamic composition of web services. Principally these are:

- Dynamically locating and integrating functionality from a large number of disparate sources via a common interoperability mechanism.
- Defining interface functionality without detailed prior knowledge of how it will be used.
- Separating the interoperability of application functionality from that of communication mechanisms.

We therefore propose that a service-oriented architecture be adopted for the integration of functionality present in a pervasive computing environment. In other words, all the processors, sensors, actuators, displays etc, present in pervasive computing environments can only interact through services. Therefore, developing the highly adaptive behaviour we would like pervasive computing environments to exhibit becomes an ad hoc service composition problem.

This approach potentially benefits from being able to exploit the momentum that has already built behind web services, in the form of standards for services and service composition and their system support in various development tools. A further advantage is that we can seamlessly combine local services and services available over the WWW in supporting a user's requirements.

A service-oriented approach to interoperability in pervasive computing presents some additional challenges. For instance, many pervasive computing devices have limited processing and memory footprints and wireless communication protocols offer differing reliability and quality of service profiles when compared to the Internet, so much of the existing Web Services infrastructure may not be directly applicable to this area. However, for the purposes of this paper we will be focusing on the challenges of automated ad hoc service composition in the pervasive computing domain. In particular we address the problems of application level interoperability and the pressures placed on the service engineering process by the need for semantics. To address this we take the view that the service-oriented approach must be supplemented by application level semantics capable of supporting machine reasoning in interactive timeframes.

3 Semantic Interoperability and Ontologies

Sheth classifies the types of interoperability problems that can occur during the interchange between information systems as follows (Sheth 1998):

- System, heterogeneous hardware and operating systems;
- Syntactic, different representation languages and data formats;
- Structural, heterogeneous model representations;
- Semantic, different meaning of terms used in the interchange.

System and Syntactic interoperability problems are more easily dealt with. However, achieving Structural and Semantic interoperability (which together will be termed

---

* We use the term 'pervasive computing' synonymously with 'ubiquitous computing' an the term 'pervasive computing environment' synonymously with 'smart space'

semantic interoperability in this paper) in information interchange between information systems of different parties continues to be a difficult problem. In order to achieve semantic interoperability in a heterogeneous environment, the meaning of the information that is interchanged has to be understood across the systems. A simple example would be where one party might call the invocation charge for a particular web service “Fee” with the currency type "US Dollars" and the other party calls the charge “Cost” and expects a "Euro” representation.

More generally, Ceri and Widom identify four categories of semantic conflicts (Ceri and Widom 1993):

- Naming conflicts where different names used to represent the same concepts, either homonyms and synonyms .
- Domain conflicts occur when different reference systems are used to measure a value. Examples are different currencies.
- Structural conflicts occur when different systems use different data organisation to represent the same concept.
- Metadata conflicts occur when concepts are represented as one type within the modeling type of one system and a different type within the other system (e.g. as at a schema level in one database and as an instance in another.

In the past, such semantic conflicts have typically been dealt with at "design time": through careful schema design in distributed database solutions; through the hand crafting of interoperability gateways (with system integrator solutions); or by forcing each system to conform to a standard mechanism for interchange (e.g. ebXML). Although these traditional approaches have been successful in well understood/static interchange environments, each of these approaches are inappropriate for systems that want to interchange in dynamic environments (Cui 2002): the schema design solution fails due to the rapidly changing nature of the interchanges required; the handcrafting of gateways solution fails as it does not scale to large numbers of information systems; and the standards solution fails due to the lack of certainty as to whether there is a common interpretation of the standard.

Ontologies can be used to describe the semantics of information sources and make the content explicit, and thus can be used to discover semantic equivalence between information concepts. The use of ontologies as a possible solution to the semantic interoperability problem has been studied over the last six or seven years. Wache et al reviewed and categorised twenty five approaches that have been proposed over this period and concluded that reasonable results for integration can be achieved through ontology based approaches (Wache 2001). Like the traditional approaches most of the approaches proposed are "design time" solutions towards integration. However as ontologies represent semantics, then semantic interoperability can be achieved by runtime comparison of and inference about ontological information.

The role of ontologies is becoming important in order to realise the vision of the Semantic Web (www.semanticweb.org) and several XML based representation languages are emerging. DAML and DAML+OIL (DAML 2003), and OWL (W3C 2003) are based on W3C's Resource Description Framework (RDF) (W3C 2000). XML Topic

---

* We use the term ‘pervasive computing’ synonymously with ‘ubiquitous computing’ an the term ‘pervasive computing environment’ synonymously with ‘smart space’

Maps (XTM 2003) are based on the ISO 13250 standard (which was originally defined to facilitate merging of indexing schemes) All of the above representation languages are seen as candidate web-based XML languages for expressing and interrelating ontologies. Obrst and Liu provide a good insight into the relationship between Knowledge Representation and Ontological Engineering, and the emergence of web based languages (Obrst and Liu 2003).

Of course in a dynamic environment such as the one that we envisage, people will have freedom of choice in terms of how they structure their semantics and the representation language that they use to do so. We will term this the ontology heterogeneity problem. Visser et al focuses on ontology hetereogeneity and classifies mismatches that may occur under headings of Conceptualisation (on how concepts are classified) and Explication (on how concepts are specified). They then go on to discuss how easy/hard it is to deal with each type (Visser 1997). More recently Klein has undertaken a similar analysis of the mismatch problems that can occur and examines how various projects propose solutions to these problems (Klein 2001). Several papers propose particular approaches or architectures for resolution of ontological heterogeneity, mainly at design time. For example Corcho and Gomez-Perez propose a system that will semi-automatically integrate ontologies of different types in the e-business domain (Corcho and Gomez-Perez 2001). With respect to runtime solutions, Campbell et al outline a number of issues that typically need to be addressed (Campbell et al 1995):

- How to find a common frame of reference in both ontologies around which algorithms can traverse?
- Identifying what is or is not a match? What constitutes an exact match or a "good enough" match and how is this expressed?
- What online resources that might help identify equivalent terms (e.g. WordNet) should be used and when?

In summary, whereas the use of ontologies has been shown to overcome the semantic interoperability problem between parties that know they want to interact a priori, the use of ontologies to achieve semantically interoperable interactions between parties at runtime in a very dynamic situation is still in its infancy and must overcome a number of challenges.

3.1 How will Ontologies help in Pervasive Computing?

Ontologies could be useful at design time to help pervasive computing engineering, in a similar way to how it has been found useful in the support of system integration at design time. With semantic markup and ontologies, knowledge about various pervasive computing parties (physical or software) can be encoded at creation time and reasoned upon later at reuse or selection time by designers of pervasive computing systems.

The pervasive computing system could also start to reason about relations between users' context and needs, available device capabilities and the physical environment at runtime. It has been argued that ontologies are "overkill" as a solution to support runtime interactions. However, Kim summarises the situation nicely that in general XML DTDs will be good enough to support runtime interactions in systems where the pressing need is reduction of complexity and common agreement amongst various parties can be

---

* We use the term ‘pervasive computing’ synonymously with ‘ubiquitous computing’ an the term ‘pervasive computing environment’ synonymously with ‘smart space’

achieved, whilst if the pressing need is reduction of uncertainty then ontologies are necessary (Kim 2002). The authors would argue that because of the kind of diversity and dynamics that is expected in pervasive computing environments, it is unlikely that there will be common agreement on one representation language or one set of terminology for the markup of all application elements. Therefore each element destined for use in a pervasive computing environment must encode knowledge about itself using explicit reference to an appropriate ontology. Thus, when such elements are integrated at design time or even at runtime for a pervasive computing system, several ontologies may need to be navigated and analysed in order to dynamically bridge understanding between the constituent elements.

4 Semantic Service Integration

As the open expression of semantics, in the form of ontologies, supports machine reasoning, there is increasing interest in exploring how they can play a role in supporting increased dynamicity in the composition of services.

The DARPA Agent Mark-up Language initiative is addressing the need for automated service composition by taking an ontology based approach to defining services and service compositions called DAML-S (damls 2002). As with other service composition languages DAML-S is bound to WSDL. However it allows for the definition of inputs and outputs to service operations to be defined in terms of ontology elements. This allows for a richer expression of the semantics of inputs and output information than is possible with the XML schema information representation used in WSDL. DAML-S also allows for the definition of logical expressions for the pre-conditions to the use of the service and for the effects of using the service, again in terms of ontology elements. In this way the semantics of the service can be expressed in terms of an ontological model of the ‘real-world’ context in which the service operates. In addition, DAML-S provides some common semantics related to the resources used by a service and how they may be shared. This is an important abstraction in managing the assurance of a service, since the quality of service it is able to deliver will depend on the sharing of such resources. DAML-S thus allows for automated reasoning about the compositions in which the service may be involved without the service designer having to explicitly address the requirements of that particular composition. This is needed in pervasive computing environments where composite service must be formed from several services with no a priori knowledge of each other (Chakraborty 2002).

DAML-S presents a promising candidate for an open language for the composition of semantic services since it is expressed in the DAML+OIL language which has heavily influenced the W3C standardisation effort for a Web Ontology Language (OWL). Future revisions of DAML-S will therefore be built on OWL. However DAML-S is still very much at the research stage, and reasoning application are targeted more at design time tools than at dynamic runtime scenarios (McIlraith 2001).

5 Roadmap

This section outlines a roadmap identifying the different scenarios in which semantic interoperability may contribute to the engineering and operation of pervasive computing systems. It takes as a starting point the assertion that all the different functionality found

---

* We use the term ‘pervasive computing’ synonymously with ‘ubiquitous computing’ an the term ‘pervasive computing environment’ synonymously with ‘smart space’

within a pervasive computing environment will be expressed as a semantic service in an ontology-based language such as DAML-S. The roadmap lays out the possible evolution of human roles involved in the composition of semantic services, specifically the user of composite services, those involved in designing composite services and those involved in developing atomic services for specific functionality they have developed.

Near Term:

Service Users: Are offered a limited set of existing (composite) service offerings, either generic or pre-tailored to frequently encountered pervasive computing situations. Automated adaptation of these compositions is very limited, addressing specific areas such as terminal adaptation, where industry agreed profiles exist.

Service Composers: Work either in developing custom compositions for specific pervasive computing environments, e.g. for corporate customers, or in contributing to open-source composition repositories. Their differentiating skills lie in their understanding of semantic service expressions and their experience in using knowledge management and service composition tools. They often have to retro-fit semantic description to required services that have not yet been marked-up.

Atomic Service Developers: Needs to use knowledge management tools to import and understand ontologies, but often also find themselves developing ontological models where no suitable ones exist. Their existing design tools are not well integrated with the knowledge management tools.

Medium Term

Service Users: Are served by dynamic service offerings selected or generated to satisfy user specifications of the task at hand, including group tasks, and of usage preferences.

Service Composers: Consists of two groups. One is skilled in optimising service compositions, especially relating to non-functional requirements. The other consists of domain specialists who construct compositions to address difficult problems for their domain community – they increasingly identify themselves as being primarily Service Users.

Atomic Service Developers: Benefit from some integration of knowledge management tools with their native design tools, so that browsing, matching and mapping between ontologies is better integrated into the design process. They may find themselves involved in developing ontology agreements for their application domain.

Long Term

Service Users: Just indicate intent through spoken word and gestures, which are understood by pervasive computing environment in concert with learnt preferences and understanding of on-going user tasks and social setting.

Service Composers: Now retired, agents perform service composition at run-time.

---

* We use the term ‘pervasive computing’ synonymously with ‘ubiquitous computing’ an the term ‘pervasive computing environment’ synonymously with ‘smart space’

Atomic Service Developers: Focuses on development of domain-specific embedded functionality, the development tool handles the generation of semantic service definitions based on knowledge gleaned from interactions with the developer during the design process.

Though elements of this roadmap may be highly speculative, it provides a useful tool in guiding research that spans the two fast moving areas of semantic service composition and pervasive computing. This paper now focuses on two relevant topics:

- Semantic interoperability, i.e the matching of meaning and its role in automated mediation between run-time systems
- Semantic Service Engineering, i.e. the motivations, process, and challenges of developing atomic and composite semantic services for pervasive computing environments.

6 Dynamic Auto-generation of Semantic Interoperability Gateways

Our group has been investigating semantic web technologies to dynamically bridge between the terminology of two parties from different pervasive computing environments which we assume have been developed using different ontologies (O'Sullivan et al 2002). First the ontologies of the two parties are compared. The comparison of the ontologies is then used to derive rulesets that transform the negotiation in one party's terminology to that of the other party's terminology and vice versa. This comparison and dynamic bridging is undertaken at runtime and requires no a priori knowledge of the parties. Key to the approach is that each ontology need only adhere to a minimum subset of common concepts, with the rest of the ontology being defined to reflect the terminology most natural for the particular pervasive computing environment. This minimum subset of common concepts approach is a variation on the "hybrid ontology approach" outlined in (Wache 2001). Of course, deriving what the minimum subset should be, is a major challenge.

An initial prototype has been developed in order to explore the issues involved in the proposed solution. As shown in Figure 1, the initial prototype uses Enterprise Java Beans and XML technologies for its implementation. The scenario chosen involved a person roaming from one pervasive computing environment to another, but where there was no Service Level Agreement in place for that person in the target environment and where the service terminology used across the environments were different. The aim was to develop a system that would allow the negotiation to take place at runtime and real-time such that seamless continuity of appropriate services were maintained. Thus requests in one domain would issue requests/responses in its own service and quality of service terminology. These requests/responses would get translated in the Semantic Interoperability Gateway into requests/responses of the receiving party's terminology and vice versa.

---

* We use the term ‘pervasive computing’ synonymously with ‘ubiquitous computing’ an the term ‘pervasive computing environment’ synonymously with ‘smart space’

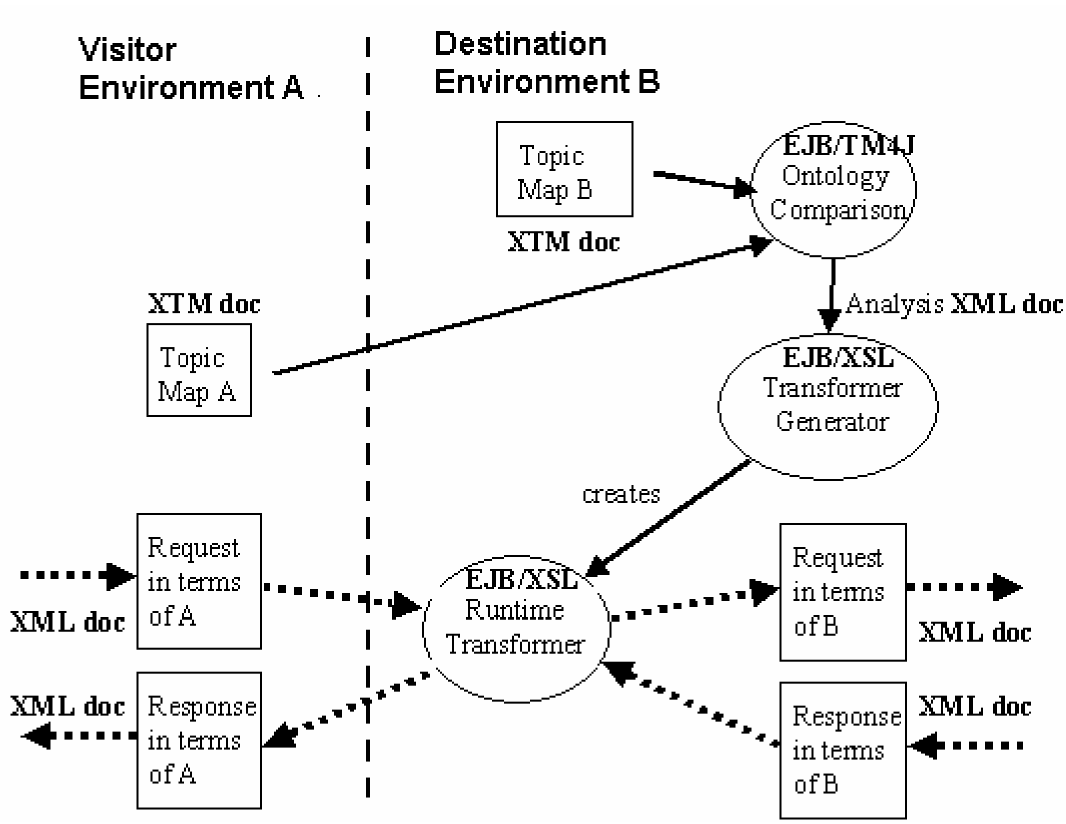

Figure 1. Prototype Auto Generation of SI gateways

The ontologies in the prototype are described using the XML Topic Map (XTM) representation (www.topicmaps.org). A subject, in its most generic sense, can be any ‘thing’ whatsoever — a person, an entity, a concept, really anything — regardless of whether it exists or has any other specific characteristics, about which anything whatsoever may be asserted by any means whatsoever. A topic is created within the topic map to ‘indicate’ (or refer to) this subject. A topic association asserts a relationship between two or more topics. A topic may be linked to one or more information resources that are deemed to be relevant to the topic in some way. Such resources are called occurrences of the topic.

A “Published Subject Indicators” (PSI) is any subject for which a subject indicator has been made available for public use and is accessible online via a URI, and is therefore any resource that has been published in order to provide a positive, unambiguous indication of the identify of a subject for the purpose of facilitating topic map interchange and mergeablity. When two topics use the same PSI to indicate their subject, they are by definition “about” the same thing.

Figure 2 provides a simplified representation of two example topic maps used in the prototype, each describing services of an independent pervasive computing environment. We are assuming that the terms which are underlined in the diagram (e.g. Service, Activity, Recall, Record, qos etc.) will be subject to common agreement on their semantics. In this case the “Recall” service term would be commonly defined as being a service that “Allows pulling up and display of information of relevance to the activity

* We use the term ‘pervasive computing’ synonymously with ‘ubiquitous computing’ an the term ‘pervasive computing environment’ synonymously with ‘smart space’

underway" (e.g. the retrieval and display of minutes where a meeting activity is underway). When both parties use the common concepts indicated when representing activities and services in their topic maps, it is possible for a software tool to analyse the topic maps to check for equivalence. Thus the implementations of service "LazyRecall" in TM1 and service "RememberSoon" in TM2 could be considered equivalent as both are of type "Recall" service and have equivalent quality of service attributes. In addition, activity "Brainstorm" in TM1 and activity "Creative Session" in TM2 could be considered equivalent as both are supported by services of equivalent types.

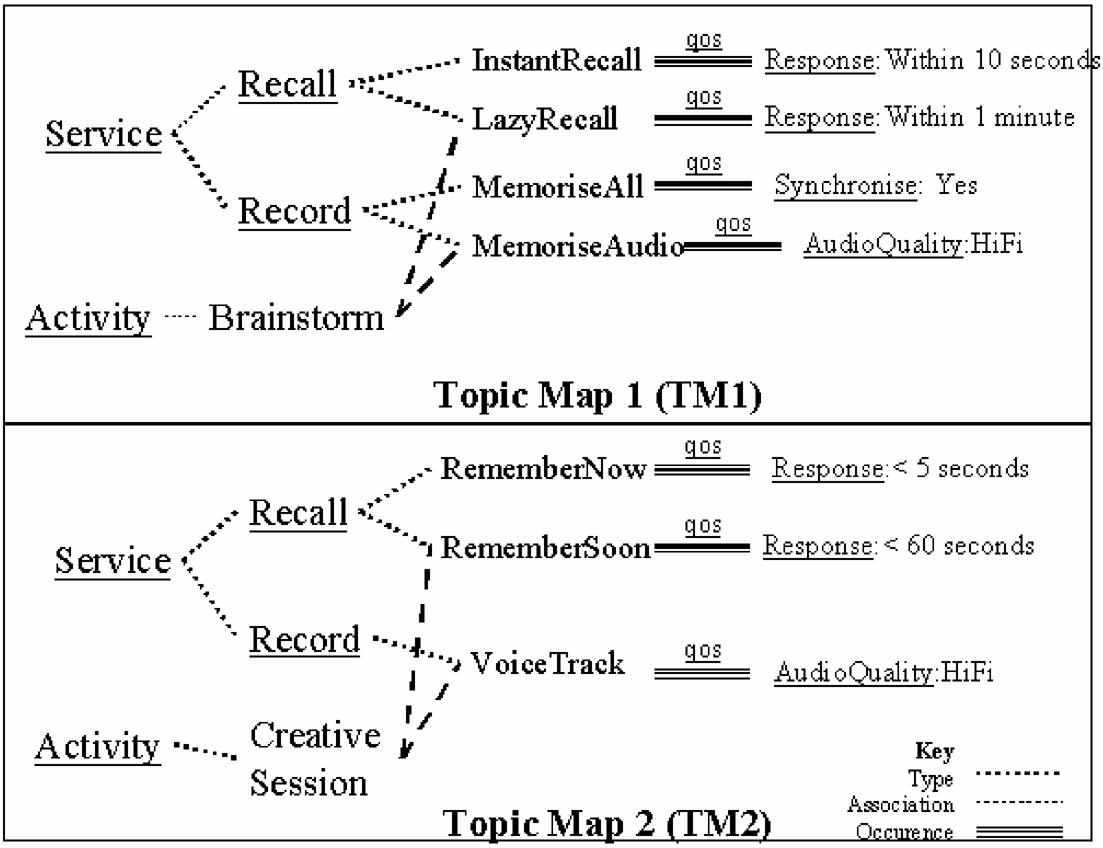

Figure 2. Example Topic Maps

Given two topic maps, the Ontology Comparison component will produce an XML document that indicates which services and activities are exactly equivalent, partially equivalent or do not match at all. The Transformation Rules generator component will take this analysis XML and create two XSL rulesets (one for each direction of communication) that can be used dynamically at runtime to transform the terms found in service requests and service responses into terms understandable by the other party. How the Transformation Rules generator interprets the analysis document (i.e. how it handles exact/partial/no matches) is determined by an XSL ruleset. In this way the behaviour of the component can be adapted easily.

Although the prototype shows that the approach holds promise, a number of questions are being pursued in the evolution of this work:

---

* We use the term 'pervasive computing' synonymously with 'ubiquitous computing' an the term 'pervasive computing environment' synonymously with 'smart space'

- How realistic is the definition of a "minimum" of common concepts to be used in pervasive computing ontologies? If not, how can equivalence between concepts be deduced?
- What effect will different ontology representations have on the performance of the comparison analysis?
- What software mediation or software adaptivity approaches are most appropriate in the construction of the semantic interoperability gateways?

7 Semantic Service Engineering

Though the above section demonstrates the potential power of ontologies in performing automated semantic interoperability at runtime, this is predicated on the availability of a large number of different services all willingly exposing their interaction semantics. To reach such a conducive environment will, as our roadmap indicates, require the cooperation of the developers of services, both atomic and composite, in accurately defining the service semantics for their products. For them to be motivated to undertake this perceived overhead, it must be both beneficial for them, by increasing the utility of their services, and easy to accomplish. Easing the process of semantic mark up for services requires integration of the software engineering process that produces the implementation of the service and the knowledge capture and maintenance process that results in usable ontologies. In a broad sense this requires an integration between the fields of knowledge engineering and software engineering. Some progress is being made towards such an integration. Based on the observation that ontology languages such as DAML+OIL share many characteristics with object-oriented model, the OMG is currently examining support for ontology modeling within the Unified Modelling Language (OMG, 2002). Some object-oriented CASE tools already offer support for exporting class models in ontology languages, albeit with proprietary mappings between the objected-oriented and ontology meta-models.

Though support for ontology languages is a necessary step for the integration of software engineering and knowledge engineering processes for service-oriented systems, it is not sufficient. The location, navigation, manipulation and processing of ontologies need to be integrated into the development process for service-oriented software development. In particular, the potential for ontologies to support machine inference about the selection and testing of services and the verification and simulation of service compositions must be exploited in easing the semantic service development process. However, as the optimization of techniques for service-related machine reasoning are still at the research stage, semantic service development tools are still largely based on knowledge management tools (McIlraith 2001) and are yet to be integrated into main stream engineering tools.

Machine reasoning has previously seen application to service engineering, principally in the telecommunications industry. Formal languages such a the Service Description Language (SDL) (Ellsberger et al 1997) were used to provide formal definition of services implemented in component-oriented telecommunication control systems, often referred to as intelligent networks. Reasoning with such formal service descriptions was

---

* We use the term 'pervasive computing' synonymously with 'ubiquitous computing' an the term 'pervasive computing environment' synonymously with 'smart space'

used initially to detect unwanted feature interactions in the service creation process for intelligent networks, but research was also conducted into using them for user evaluation trials, design simulations and automation of test case generation (Lodge 1999). SDL bears many similarities to DAML-S, as both express logical assertions about preconditions and post-conditions, i.e. effects, of service invocation, and can also express logical relationships and constraints between service inputs and output. However, the use of formal reasoning in intelligent network service creation was often seen as an addition to the service creation process, and therefore an overhead the benefits of which had to be directly assessed against the equivalent costs of traditional software engineering processes. Also, the logical concepts used in defining formal assertions for a particular intelligent network service was tied to that particular service and offered little benefit outside the development of that service. At best such formally defined objects were restricted to reuse in the intelligent network software market which was dominated by a few large players, and thus not open to the utility of formal definitions as a commodity or shared resource.

An ontology based approach to defining formal service semantics offers a more sustainable model for investing in the development of formal semantics than could ever have been the case for telecommunications service creation. The ubiquity of the WWW and the low entry costs to providing web services has provided a strong motivation of communities to collaborate in developing and freely publishing ontologies for their application domains in order to overcome the current semantic-based hurdle to cheap, widespread interoperability. Service developers are therefore likely to benefit from an ever growing reservoir of freely available ontologies providing logical concepts and related inference rules on a wide range of subjects. At the same time, the growing number of web services will force increased automation in their discovery, composition and invocation, such that services without well designed formal semantics will become increasingly less appealing to potential users. Service developers, therefore, will be increasingly motivated to contribute to the definition of reusable semantics in the form of ontologies. Previously, in the context of component-oriented communication management software development, we have argued that explicit publication of the information content of services eases the service integration process and the need for interoperability promotes convergence of such models over time (Lewis, 2003). We expect to see a similar effect in the development of semantic services, where freely available models of information used in existing services, i.e. ontologies, will prompt their use in new services, easing the semantic interoperability problem over time. If our assertion that an interoperability problem similar to that addressed by semantic web services will face pervasive computing, then developers of pervasive computing devices will have the same benefits of ontology reuse and the same motivations in contributing to ontology development.

8 Conclusions and Future Work

In this paper we have argued that the development of pervasive computing environments faces an interoperability challenge similar to that facing machine to machine interaction on the Web. We assert that using interoperability standards is not sufficiently scalable an approach to this problem, and that the semantic service approach being proposed for the Web is an appropriate route to a suitable adaptive solution. We outline a solution to

---

* We use the term ‘pervasive computing’ synonymously with ‘ubiquitous computing’ an the term ‘pervasive computing environment’ synonymously with ‘smart space’

dynamic interoperability based on ontologies, presenting some initial results using topic maps. We also discuss the potential challenges facing the developers of semantic services, including the need for integrated software and knowledge engineering tools and the harnessing of existing ontologies in developing semantic service descriptions and promoting semantic convergence.

Recent work has shown that expression of service capabilities in DAML-S facilitates service matching (Paolucci et al 2002). This approach compares the inputs/outputs of the required service with the inputs/outputs of the actual services available. It relies upon the availability of a common ontology for the inputs/outputs in order to facilitate matching. We plan to build upon this work, our experience of using Topic Maps, and the emergence of OWL to propose a solution that will enable service matching to take place in scenarios where heterogeneous ontologies are involved, which as stated earlier we believe will be the dominant situation. We hope to be able to assess if the DAML-S ontology provides the core common semantic needed for efficiency in dynamic semantic interoperability or to identify any further ontological concepts that are required.

More broadly our group is undertaking research into the following aspects of the application of semantic service to pervasive computing.

- Driving semantic service compositions from a combination of user intent inferred from multi-model sensors and matching inferred user tasks to existing service composition patterns
- Generating ontological overlay onto context information for pervasive computing environments to ease its use by semantic services
- The use of policy-based management to optimise the behaviour of running service compositions both to non-functional service requirements and system administration goals.

In support of this research, and to further explore the issues of semantic service engineering, we are planning a platform for the rapid design and evaluation of atomic and composite services. This will be based on an existing reflective object-oriented application prototyping platform that will be extended to support ontological models and integrated with appropriate reasoning tools in an extensible manner. Overall, though we see semantic services supporting increasing automation of service composition that involve little or no human intervention, we recognise and actively address the need to support the development and reuse of ontologies in the human-led service development and composition process. Only such a dual approach will spur the widespread deployment of semantic services needed for automated composition while also using open ontologies to keep semantic divergence under control and thus aid automatic semantic interoperability.

Acknowledgements

This work was funded by the Irish Higher Education Authority under the M-Zones programme.

References

---

* We use the term ‘pervasive computing’ synonymously with ‘ubiquitous computing’ an the term ‘pervasive computing environment’ synonymously with ‘smart space’

Van der Aalst, W.M.P. (2003), “Don’t go with the Flow, Web services composition standards exposed”, IEEE Intelligent Systems, vol18, iss 1., Jan/Feb.

Campbell et al (1995), Campbell A, Chalupsky H, and Shapiro S. "Ontological mediation: An analysis." Unpublished manuscript

Ceri and Widom (1993), Ceri S., Widom J., "Managing Semantic Heterogeneity with Production Rules and Persistent Queues", Proceedings of the 19th VLDB Conference, Dublin, Ireland, pp. 108-119.

Chakraborty, D., Perich, F., Joshi, A., Finin, T., Yesha, Y. (2002), ‘A Reactive Service Composition Architecture for Pervasive Computing Environments’, Technical Report TR-CS-02-03, Computer Science and Electrical Engineering, University of Maryland Baltimore County

Christensen, E., Curbera, F., Meredith, G., Weerawarana, S. (2002), “Web Services Description Language (WSDL) 1.1”, W3C Note 15 March 2001, http://www.w3.org/TR/wsdl

Corcho and Gomez-Perez (2001), Corcho, O. and Gomez-Perez, A., "Solving Integration Problems of E-commerce Standards and Initiatives through Ontological Mappings", In: Proceedings of the Workshop on E-Business and Intelligent Web at the Seventeenth International Joint Conference on Artificial Intelligence (IJCAI2001) , Seattle, USA, August 5, 2001.

Cui (2002), Cui Z., Jones D. and O'Brien P., "Semantic B2B integration: issues in ontology-based approaches", ACM SIGMOD Record, vol 31 no 1, pages 43—48, 2002, ACM Press

DAML (2003), DARPA Markup Language, www.daml.org, Visited Apr 2003

Ellsberger, J., Hogrefe, D., Sarma, A. (1997), “SDL, Formal Object-oriented language for communicating systems”, Prentice-Hall

Goh (1997), Cheng Hian Goh, "Representing and Reasoning about Semantic Conficts in Heterogeneous Information Sources", PhD, MIT, 1997.

Klein (2001) Klein M. "Combining and relating ontologies: an analysis of problems and solutions". In IJCAI-2001 Workshop on Ontologies and Information Sharing, pages 53--62, Seattle, WA, 2001.

Lewis, D., Wade, V., Cullen, B. (2003) “Towards the Technology Neutral Modelling of Management Components”, Journal of Network and Systems Management, vol 11, no. 1, March

Lodge, F., Kimbler, K., Hubert, M. (1999) “Alignment of the TOSCA and SCREEN Approaches to Service Creation” in Proceedings of the 6th International Conference on Intelligence in Services and Networks, Barcelona, Spain, Springer-Verlag, pp277-290

Obrst and Liu (2003), Obrst L. and Liu H., "Knowledge Representation, Ontological Engineering and Topic Maps", Chapter 7, XML Topic Maps, Jack Park (ed.), Addison-Wesley.

OMG (2002) “UML Profile for Web Ontology Definition, Request for Proposals”, OMG document ontology/2003-12-03.

O'Sullivan D., Wade V., Lewis D., Conlan O. (2002), "Enabling Smart Space Service Mobility Negotiation by Ordinary Users", Proceedings of EURESCOM Summit 2002 Heidleberg, Germany

* We use the term ‘pervasive computing’ synonymously with ‘ubiquitous computing’ an the term ‘pervasive computing environment’ synonymously with ‘smart space’

McIlraith, S.A., Son, T.C., Honglei Zeng, H. (2001), ‘Semantic Web Services’, IEEE Intelligent Systems, 16(2), March/April 2001

Paolucci M., Kawamura T., Payne T, and Sycara K (2002) .,"Semantic matching of web services capabilities", ISWC2002, 2002.

Sheth (1998) Sheth A.P. , "Changing Focus on lnteroperability in Information Systems: From System. Syntax, Structure to Semantics", M. F. Goodchild, M. J. Egenhofer, R. Fegeas, and C. A. Kottman (eds.) Interoperating Geographic Information Systems, Kluwer.

Visser (1997), Visser, P.R.S., Jones, D.M., Bench-Capon, T.J.M., and Shave, M.J.R., "An analysis of ontology mismatches; heterogeneity versus interoperability." AAAI 1997 Spring Symposium on Ontology Engineering, 1997.

Wache (2001), H. Wache, T. Vögele, U. Visser, H. Stuckenschmidt, G. Schuster, H. Neumann, and S. Hübner, "Ontology-based Integration of Information - A Survey of Existing Approaches," In: Proceedings of IJCAI-01 Workshop: Ontologies and Information Sharing, Seattle, WA, 2001, Vol. pp. 108-117.

W3C (1999) World Wide Web Consortium, Resource Description Framework (RDF) Model and Syntax Specification, W3C Recommendation, http://www.w3c.org/, February 1999.

W3C (2003) Ontology Web Language, http://www.w3.org/2001/sw/, Visited Apr 2003.

Weiser, M. (1991), “The Computer of the 21st Century”, Scientific American, vol. 265, no.3, September 1991, pp 66-75.

XTM (2003) XTM Authoring Group, XTM: XML Topic Maps 1.0. www.topicmaps.org. Visited Apr 2003

‘DAML-S: Semantic Markup for Web Services’, The DAML Service Coalition, http://www.daml.org/services/, October 2002.

---

* We use the term ‘pervasive computing’ synonymously with ‘ubiquitous computing’ an the term ‘pervasive computing environment’ synonymously with ‘smart space’