Managed Person-centric Adaptive Services for Smart Spaces

David LEWIS, Aoife BRADY, Kevin CAREY, Owen CONLAN, Kevin FEENEY, Steffen HIGEL, Tony O'DONNELL, Declan O'SULLIVAN, Karl QUINN, Vincent WADE
Knowledge and Data Engineering Group, Computer Science Department, Trinity College Dublin, Dublin 2, Ireland
Tel: +353 1 6082158, Fax: +353 1 6772204 , Email: Dave.Lewis@cs.tcd.ie

Abstract: This paper outlines the requirements for an adaptive service-oriented architecture for ubiquitous computing. We establish the need for the architecture to support natural expression of user requirements and constraints for adaptive behaviour and detecting and reacting to changes in context. We adopt service composition and policy-based management as the primary adaptive mechanisms. We exploit ontology languages to support both an open engineering approach and the automated reasoning needed for runtime adaptation. An abstract model of the architecture is described and current work towards populating the architecture is outlined.

1. Introduction

The commonly articulated vision of ubiquitous computing [24], where processors, sensors, actuators and displays are integrated into the fabric of everyday life, represents a huge increase in the number of independently developed components that must interoperate. This motivates a shift to exchanging interoperability knowledge between components at runtime rather than between humans at development time. This shift is coupled with the ability of components to collectively adapt themselves dynamically to the requirements of users transferring between physical spaces and tasks over time. The confluence of the service-oriented techniques and ontology-based semantics as Semantic Web Services [16], offers dynamic adaptivity through knowledge-based service composition. In [17], we observe that the likely range of services that must be integrated in any particular situation will require both the engineering discipline to provide machine-processable semantics for deployed services and the dynamic generation of interoperability gateways. In this paper we discuss an architecture for the knowledge management and reasoning infrastructure that can efficiently support both the needs of dynamic semantic service composition and the management of the resulting adaptive services.

Consider the scenario of a university student coming to meet a lecturer for a tutorial session (see figure 1). Both the student and the lecturer may bring their own computing resources to the meeting, e.g. laptops and PDAs. They may also, via wireless networks, have access to resources in the locale such as printers, file servers and desktop computers. Individual resources are made available through adaptive software components. These are used by the adaptive system which implements the ubiquitous computing environment, to dynamically generate the services needed by the student and the lecturer, e.g. a service for minuting the meeting. In this setting, sensed context such as the identities of the people in the room, the resources available and the recognition of acts, such as spoken commands or gestures, could all be made available to the adaptive system when determining the behaviour of the services that should be offered to the users at any point in time [7]. However, this adaptive process should also be informed by default behaviour rules contained in personal preferences, as well as restrictions on what the people involved are permitted to do, e.g. whether the student send a document to the lecture's local printer. The latter may involve policies that are derived from the wider organizational setting of the university's bureaucratic organization, e.g. an information services committee that sets wireless LAN usage policies. Such a scenario highlights the need for the adaptive system to mediate between, sometimes imprecise, expressions of individual intent, of group tasks and organization-level policies and of the adaptive mechanisms that generate and manage the tailored services provided to the user.

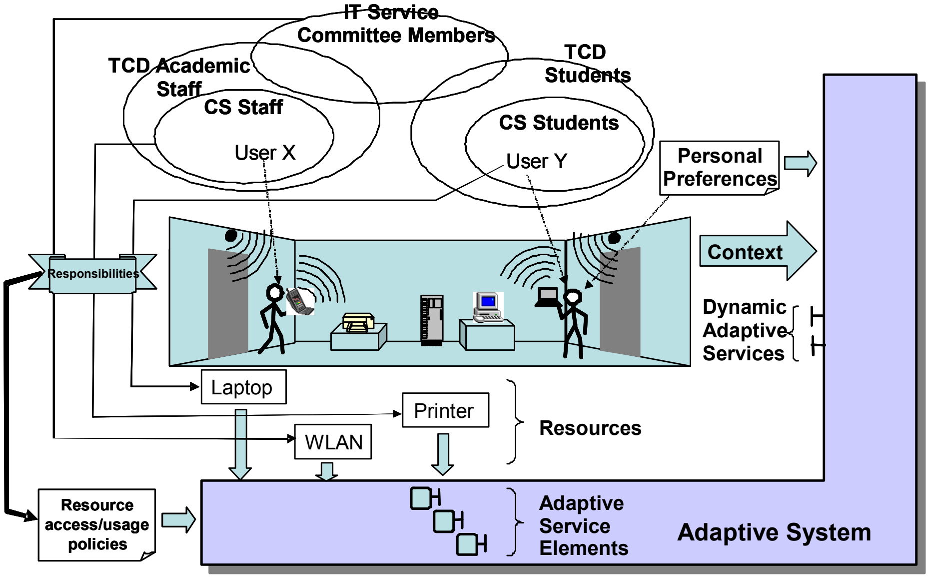

Figure 1: Example of adaptive system operation in a ubiquitous computing environment

This example illustrate on the need to integrate service adaptivity with the management of services elements and resources that constrains that adaptivity. Both need to operate within an architecture that supports the heterogeneity and dynamicity of systems. While adaptivity is of obvious benefit to individual end users, users working in communities or organisations are the beneficiaries of operational constraints on adaptivity as this provides the means for managing the resources and services for which they have operational responsibility. Being able to dynamically handle heterogeneity is of benefit to organisation operating smart spaces, while the architecture also support service component and device vendors develop adaptive ready products for the broadest possible markets.

2. Background

This section outlines the main techniques and technologies upon which the architecture is based. Ubiquitous computing environments will exhibit a large amount of heterogeneity in the components from which they are constructed. Any adaptive system supporting ubiquitous computing will therefore face major interoperability and integration challenges in combining the adaptivity of user services with adaptive resource management. These problems can be eased by adopting a service-oriented architecture in which constituent elements are accessed via a well-defined, self-describing interface.

Adopting a service-oriented architecture, involves constructing system functionality as adaptive service elements which can be dynamically assembled to build new adaptive services that satisfy immediate user task requirements. Elements interact though well defined service interfaces, allowing a ubiquitous computing environment to be constructed from elements sourced from any number of developers. Service-oriented architectures are inherently flexible, with system adaptability being achieved by deploying and using services in different combinations, a process known as *service composition* [1]. However, research into service composition has tended to focus on the composition mechanism rather than on guiding composition to empower the user to do want they want in the manner they want to do it.

There is increasing interest in automating the service composition process, so that the service offered to users appears to be adaptive, i.e. the service offered changes automatically according to the task the user wishes to perform and the context in which they wish to perform it [20]. However, service-oriented architectures tell us little about how such adaptive services can be used to allow people to interact with a ubiquitous computing environment in a seamless and unobtrusive manner. This requires us to go beyond existing approaches to automatic service composition, which are usually driven by some technical specification of the overall service required, to specify the required composite service in a manner more readily generated by user needs and their current context.

Another adaptive technique that is seeing increased deployment in network and system management is *policy-based management* [21]. It uses expressive rule languages to determine behavioural rules for how a system should respond to predetermined events and system conditions. Though policy-languages have been developed that can express policies at a relatively high level of abstraction [4][12][23], automatically mapping these to rules that can operate on heterogeneous, system-level resources is problematic [10]. Such mapping, together with handling the policy rule conflicts that inevitably arise on any non-trivial scale system, typically requires expert understanding of both the goals to be satisfied and the semantics of the resources used to achieve those goals [5][14]. In ubiquitous computing environments, anyone entering the space may share resources they possess or use shared resources already in situ. Policies provide a way of managing such ad hoc collections of resources, but need to employ flexible means of binding resources and policy subjects to rules at runtime [12].

Ontologies have emerged from the knowledge-engineering community as a way of expressing, merging and reasoning over agreements on domain semantic established by heterogeneous groups. The standardisation of RDF for basic subject-predicate-object expressions in XML has formed the basis for the Web Ontology Language [18] which enables web content to be semantically marked up. Further consensus is being reached on ontology based languages for services [19] and rules [9], thus providing the opportunity for applying ontology based reasoning to service composition [15] and policy-based management [22]. These languages and their processing by general purpose reasoners make ontologies a key technology in developing open engineering approaches to support automated service adaptation and interoperability.

3. Adaptive Service Architecture

Our approach to developing, operating and managing ubiquitous computing environments is based on an evolving, abstract model of an adaptive system. This adaptive system model does not address all the self-management considerations of a fully autonomic system [13]. However, it does focus on the capture of naturally expressed user control and management requirements and their automatic mapping onto adaptive mechanisms. Thus this adaptive system model could form part of a broader autonomic system architecture. The adaptive system model is based on the assumption that all functionality in a ubiquitous computing environment that is availed of by users (or their agents) is provided via services. A service provides access to a specific set of resources. Examples could be a service that allows the resources of a printer to be used to print documents, or a service that uses the resources of a data projector to display application interfaces. Resources are controlled by the implementation of the service, either solely or shared with implementations of other services. Ideally, services should represent the only way in which these resources can be manipulated via a computing system, though backward compatibility issues may occasionally prevent this. The binding between services and resources is static, i.e. the resources used dictate the nature of the service.

Service-oriented architectures are becoming increasingly common, especially with the popularity of web services that use SOAP, WSDL and UDDI infrastructure. However, when applied to ubiquitous computing environments, we have a greater need for services to autonomously adapt their behaviour rather than being adapted by the action of a human developer or administrator. More specifically, services must adapt their behaviour in response to both changes in their operational context and changes in the condition of the resources handled by the service. In practice, the implementation of a service may make use of other services, so that the service's behaviour will include the definition of when these other services are invoked and how they are used.

We model a service and its behaviour using the abstract concept of an Adaptive Service Element (ASE). This offers a specific service, the behaviour of which:

- Is aware of context information that we assume has been made available in the ubiquitous computing environment;
- Controls and is aware of the state of specific resources;
- May involve use of other services.

We envisage that such ASEs will range from specific software implementations to elements that are automatically created and deleted on the fly, e.g. ones that are compositions of other existing services. In all cases, however, the adaptive behaviour of an ASE may need to be managed to reflect the goals and preferences of both the users using the service and the people responsible for the resources which the service uses. This management is performed by providing behavioural rules to the adaptive service element. These rules dictate the element's behaviour within the constraints provided by the element's developers, be they human designers or automated agents that generate service compositions.

Given the need to generate behavioural rules, we are presented with two major interoperability challenges. The first occurs when adaptive systems attempt to automatically generate the behavioural rules based on the user model and the context of the task-at-hand. The second occurs when coordinating behavioural rules that may be understood by separate ASEs from different sources. In both cases mappings need to be established between the semantics of the rule constraints of different ASEs and the semantics of the behaviour the system as a whole.

Figure 2 outlines how we expect adaptive behaviour to be governed. On the left-hand side we see how the architecture deals with per-task adaptivity, inferring the user intent from sensed user behaviour and transforming this to a service request that is dynamically fulfilled by the generation of a composite service. On the right hand side we see that user-level behavioural rules, expressed both as individual personal preferences and as organizational policies, need to be resolved into behavioural rules applicable to composite services. These composite service level rules must, in turn, be enforced through decomposition into rules that can be applied locally to the individual application service elements that make up the composite service. These user-level rules need to be expressed in terms to which the user can relate. In particular they should be expressed in terms that relate to the tasks the user wishes to perform and the effectiveness or quality of service they expect from the adaptive application generated by the ubiquitous computing environment to support this task [11]. These personal policies need to be effectively resolved onto system level policies and reconciled with the policies set by other users, teams and administrators responsible for resources they happen to be using. Mapping individual and organizational user-level policies to system level policies presents a challenge in ubiquitous computing environments as these will often be supporting fluid, collaborative organizational structures with distributed, overlapping responsibilities for authoring policies on resources.

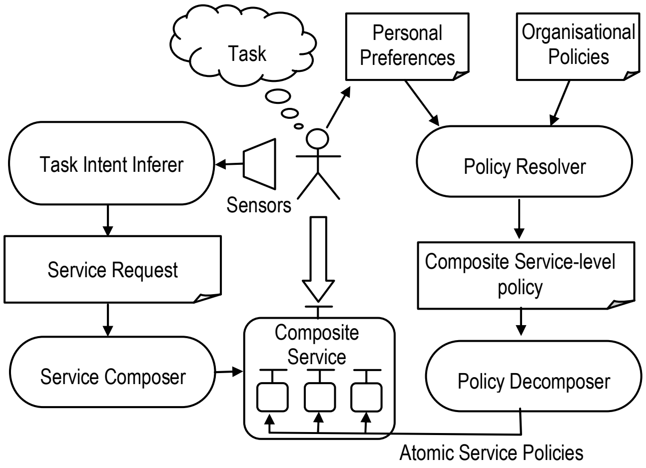

Figure 2: Integration of adaptive service composition and adaptive policy-based management

To enable the adaptive system to process such behavioural semantics automatically we adopt ontology-based semantics as a means of describing constraints on an adaptive service element's behavioural rules in a machine intelligible form. The expression of behavioural constraints is supported by having the semantics of services and the operational context expressed in an ontological format. Ontologies for service specifications are already emerging under the semantic web community [19], which promise the automation of service discovery and composition [16]. Such semantic services are also highly applicable to the ad hoc, dynamic service composition needed for ubiquitous computing [15][2], as ontologies enable a more open corpus of service inputs, outputs, precondition and effects. Ontologies will, therefore, provide an extensible and flexible way of expressing the basic terms that will make up the behavioural rule vocabulary for an adaptive service element [22]. However, the issue raised by the heterogeneity of ontologies and how to achieve semantic interoperability between systems using different ontologies remains a challenge [17].

As the ASE is a core component of our architecture we examine it here in more detail (see figure 3). An ASE is characterized by:

- a Service Description,
- a model of the state observable by the ASE, i.e. the State View;
- a description of the services of which it makes use, i.e. the Utilised Service Model;
- a rule-based model for describing and restricting its behaviour.

The lifecycle of an ASE is managed primarily through the bindings made between these models.

Service Descriptions are specified in OWL-S language [19] where a service is described using a description logic ontology specifying inputs, outputs, preconditions and effects. Compatible with the OWL-S service description, the ASE state view is an ontological model of the objects of which the element is aware, including managed resources, external context and operational state such as counters and timers.

An ASE's behaviour rules are in the form event-conditions-actions and dictate the behaviour of the service in reacting to: the service's invocation, the access control policies of the service provider, the resource management policies of the resource owner and changes to state objects. Meta-rules, typically established by the ASE developer, restrict how available events, conditions and actions can be constructed into behaviour rules, thus restricting unwanted rule behaviour. ASEs may be generated by automated service composers and thus have entirely rule based behaviour, or they may be pre-implemented software components with restricted rule-based behaviour for flexibly enforcing policies to invoke external services, such as accounting or fault management, when specific events occur.

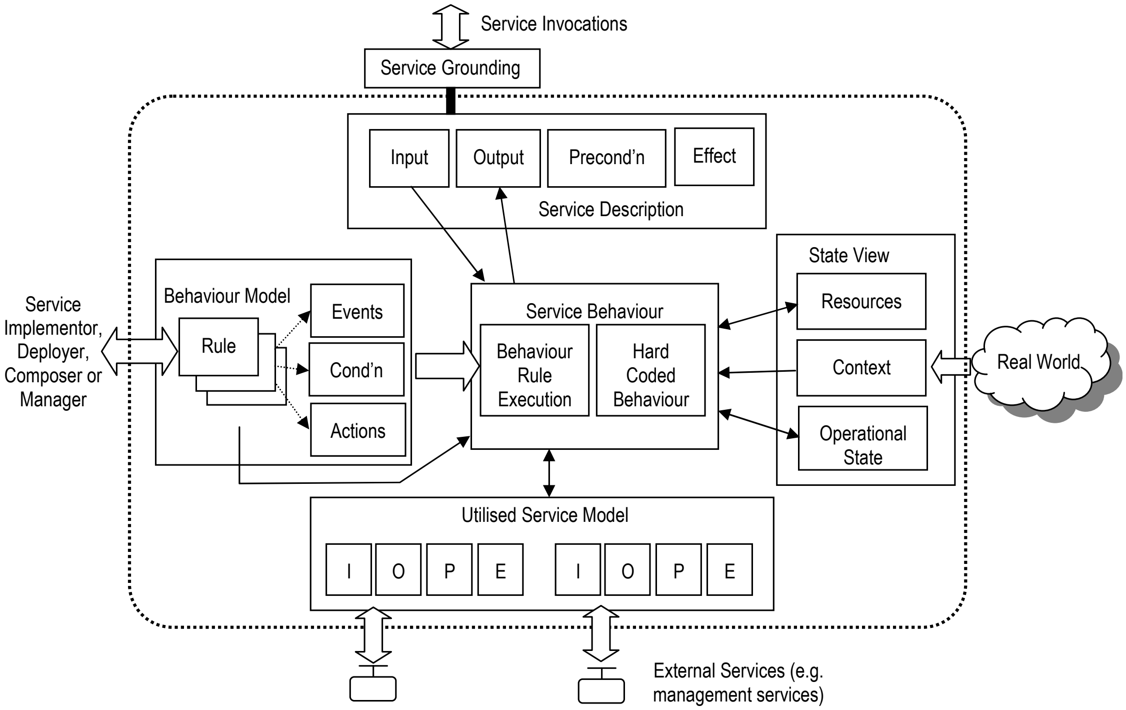

Figure 3: Anatomy of an Adaptive Service Element

4. Realisation of the Architecture

There are several hurdles that must be negotiated before this architecture can be realised in am operational environment. On the ASE supply side, industrial agreement needs to be reached by industry on a standards-based ontology, i.e. one that uses OWL and OWL-S, for representing devices and other smart space service as ASEs. This needs to be married with grounding models that tie ontological ASE specification to concrete implementation technologies. Our group is working toward the definition of such an ontology and a clear articulation of its importance in a future marketplace for managed smart space components. This involves exercising the architecture, and its prototype ASE ontology, in a number of person-centric adaptive smart space management mechanisms, so that the value-add of the approach is demonstrated and the architecture is exercised and refined.

Considering the role of the smart space operator we are examining the benefits of using an ontological model of the smart space environment and the tasks types supported by the environment to populate a Bayesian Network that is used to predict user intent based on parsing streams of sensor information [8]. These ontologies can be related to those used by available ASE's in resolving the inferred task request using automatic service composition. The use of AI planning techniques for the automatic composition of semantically-marked up services does not typically provide much manually control over the composite service delivery. We are therefore also investigating the use of human verified composition patterns in the service composition process such that composition patterns with certain desirable non-functional features are accommodated. We are also investigating the interaction between humans and the management of smart spaces using policy-based management (PBM). One problems faced by existing PBM techniques is that there will be large heterogeneity of managed resources and their usage will be combined dynamically in multiple different configurations through automated service composition. However, we are examining the use of knowledge of the resulting service composition, expressed using the OWL-S ontology, to provide automated reasoning support in mapping user level policies for service behaviour onto policy rules appropriate to the ASEs invoked in any particular service composition. Further problems arise when conflicting policies applicable to a particular ASE are authored by different groups across an organisation, e.g. IT service committee and the academic staff group from the example in section 1. Such policy conflicts may often reflect problems with the organisational structure, e.g. its decomposition into project teams or business divisions, and delegation of authority across that structure, so was are also investigating the interaction between organisational structure and the detection and resolution of policy conflicts [6].

As well as addressing the interaction between people and the adaptive systems it supports, realisation of this architecture requires investigation of certain infrastructural issues. One is the general issue of semantic interoperability, i.e. using the description logic reasoning capabilities of OWL to automatically generate transforms. We are currently investigating metrics for assessing the level to which two ontological service models are amenable to automatic semantic interoperability, so that as to guide designers of such services [17]. Another infrastructural requirement is the ability to gather context information from a variety of heterogeneous sources, e.g. sensor networks, PDAs, databases, network element agents. To support this we are investigating how ASE semantic descriptions can be used in adaptively resolving queries both originating from ASEs and served by ASEs.

5. Conclusions

This paper proposes an architecture for adaptive, managed software system for ubiquitous computing environment. This architecture makes strong use of ontologies to provide rich descriptions of adaptive service elements. The use of an ontology for representing ASE interactions provides the possibility of using a range of AI techniques to delivery adaptive person centric behaviour to smart space users, while providing the operators of smart spaces with the means to constrain that adaptivity and manage the resources employed in user services. The focus on the ASE as a component model also supports the emerging market in interoperable smart space devices by allowing some freedom in the interoperability models used and relying of semantic interoperability to provide seamless inter-working in areas where common standards may be slow to emerge.

References

[1] Chakraborty, D., Joshi, A., “Dynamic Service Composition: State-of-the-Art and Research Directions”, Technical Report TR-CS-01-19, Department of Computer Science and Electrical Engineering, University of Maryland, 19th December 2001

[2] Chakraborty, D., Perich, F., Joshi, A., Finin, T., Yesha, Y., "A Reactive Service Composition Architecture for Pervasive Computing Environments", In 7th Personal Wireless Communications Conference (PWC 2002). Singapore. October 2002.

[4] Damianou, N., Dulay, N., Lupu, E., Sloman, M., “The Ponder Policy Specification Language”, Proc. Policy 2001: Workshop on Policies for Distributed Systems and Networks, Bristol, UK, 29-31 Jan. 2001, Springer-Verlag LNCS 1995, pp. 17-28

[5] Dunlop, N., Indulska, J., Raymond, K., “Methods for Conflict Resolution in Policy-based Management Systems”, IEEE 7th International Enterprise Distributed Object Computing Conference (EDOC'03), September 16 - 19, 2003, Brisbane, Australia, pp 98-111.

[6] Feeney, K., Lewis, D., Wade, V., “Policy-based Management for Internet Communities”, to be published in the proceedings of IEEE 5th International Workshop on Policies for Distributed Systems and Networks, Yorktown Height, USA, June 2004.

[7] Hampapur A., Senio, A, Pankanti, S., Tian, Y., Pingali, G., Bolle R., “Autonomic User Interface”, IBM Research Report, RC22542, August 2002.

[8] Higel S., O'Donnell T., Lewis D., Wade V., “Towards an Intuitive Interface for Tailored Service Compositions” Porc of DAIS 2003, Lecture Notes in Computer Science 2893, pp. 266-273, International Federation for Information Processing, 2003

[9] Horrocks, I., Patel-Schneider, P.F., “A proposal for an owl rules language”. In Proc. of the Thirteenth International World Wide Web Conference (WWW 2004). ACM, 2004. To appear

[10] Hull, R., Kumar, B., Lieuwen, D., “Towards federated Policy Management”, IEEE 4th International Workshop on Policies for Distributed Systems and Networks, June 04 - 06, 2003, Lake Como, Italy, pp 183-196

[11] Hull, R., Kumar, B., Lieuwen, D., Patel-Schneider, P., Sahuguet , A., Varadarajan, S., Vyas, A., “Policy-based System for Personalized and Privacy-conscious User Data Sharing”, Bell Labs Technical Memorandum, February, 2003.

[12] Kagal, L., Finin, T., Anupam J., “A Policy Language for Pervasive Systems”, Fourth IEEE International Workshop on Policies for Distributed Systems and Networks, Lake Como, 4-6 June, 2003

[13] Kephart, J., Chess, D., “The Vision of Autonomic Computing”, IEEE Computer, Jan 2003, pp 41-50.

[14] Lupu, E.C, Sloman, M., “Conflicts in Policy-Based Distributed Systems Management”, IEEE Transactions on software engineering, vol. 25, no. 6, November 1999. pp 852-69.

[15] Masuoka, R., Labrou, Y., Parsia, B., Sirin, E., “Ontology-Enables Pervasive Computing Applications”, IEEE Intelligent Systems, Sept/Oct 2003, pp 68-72

[16] McIlraith, S.A., Son, T.C., Honglei Zeng, H., “Semantic Web Services”, IEEE Intelligent Systems, 16(2), March/April 2001

[17] 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

[18] OWL Guide, W3C, http://www.w3.org/TR/owl-guide/

[19] “OWL-S: Semantic Markup for Web Services”, The DAML Service Coalition, http://www.daml.org/services/, October 2002

[20] Ponnekanti, S.R., Fox, A., “SWORD: A Developer Toolkit for Web Service Composition”, To appear in The Eleventh World Wide Web Conference (Web Engineering Track), Honolulu, Hawaii, May 7-11, 2002

[21] Sloman, M., Lupu, E. “Security and Management Policy Specification”, IEEE Network, Vol. 16, no. 2, 2002, p 10-19.

[22] Tonti, G., Bradshaw, J.M., Jeffers, R., Montanari, R., Suri1, N., Uszok, A., “Semantic Web Languages for Policy Representation and Reasoning: A Comparison of KAoS, Rei, and Ponder" Proceedings of 2nd International Semantic Web Conference (ISWC2003), October 20-23, 2003, Sanibel Island, Florida, USA

[23] Uszok, A., Bradshaw, J., Jeffers, R., Suri, N., Hayes, P., Breedy, M., Bunch, L., Johnson, M., Kulkari, S., Lott, J., “KAoS Policy and Domain Services: Toward a Description-Logic Approach to Policy Representation, Deconflictions, and Enforcement”, IEEE 4th International Workshop on Policies for Distributed Systems and Networks, June 04 - 06, 2003, Lake Como, Italy, pp 93-98

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