

# Managing User-centric Adaptive Services for Pervasive Computing

Dave Lewis, Tony O'Donnell, Kevin Feeney, Aoife Brady, Vincent Wade

*Knowledge and Data Engineering Group*

*Computer Science Department, Trinity College Dublin, Ireland*

*{Dave.Lewis | Tony.Odonnell | Kevin.Feeney | Aoife.Brady | Vincent.Wade}@cs.tcd.ie*

## Abstract

*Pervasive computing environments need to exhibit highly adaptive behavior to meet the changing task requirements and operational context of visiting mobile users. However this must be balanced with the need of resource owners to meet their goals in administering how users use their resources. This presents challenges of how to manage adaptive systems and how such management should be exercised by people, both average pervasive computing users and administrators of pervasive computing resources.*

*This paper presents some of the issues involved in reconciling dynamic user-centric adaptation with the management of autonomic systems to meet high-level management policies. It discusses our architectural approach and presents some initial research results in addressing these issues.*

## 1. Introduction

Autonomic systems adapt their behavior to changes in user policies, context and resources [13]. Pervasive computing environments are commonly regarded as being made up of a multitude of autonomous elements collaborating to sense and respond to a user's requirements and the context of the task-at-hand [25]. We can therefore consider a pervasive computing environment as an autonomic system that is aware of a user's task requirements and operational context, the resources available to support user tasks as well as the broader management policies that also act on those resources.

In this paper, we examine some of the existing adaptive mechanisms that we see as useful for pervasive computing and then focus on some of the problems faced when ensuring these mechanisms directly support user needs. In particular, we examine how users can more naturally specify the behavior they require from an autonomic system. We consider this on both a task-by-task basis, i.e. service adaptation, and applied over all user-system interactions, i.e. adaptive management. We look at this both from the point of view of an individual user attempting to perform a

specific task, and from the viewpoint of users working collectively in organizations or communities. In this paper we focus on our architectural assumptions and on how the users, individually or collectively, can interact with these mechanisms in a way that is natural for them. Such natural interaction must address:

- How the user can easily communicate the service behavior they require at any particular point in time?
- How the user and the wider community can exert their responsibilities over any resources that may be adaptively allocated to service usage?

These requirements must balance the need to hide the complexities of adaptive service operation, while at the same time providing suitable levers of control and windows of inspection through which users can maintain a sense of ownership over adaptive service behavior. With a system that dynamically offers highly adaptive service to users, we view the management task of enforcing responsibilities over resources as essentially one of constraining the range of adaptivity the system can exhibit in different situations.

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 autonomic system which implements the pervasive 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 autonomic system when determining the behavior 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 behavior 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 autonomic 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.

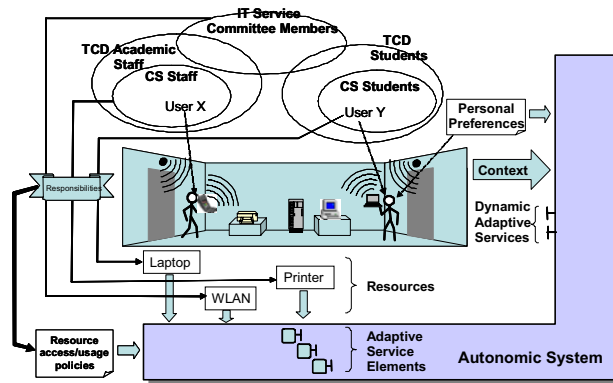


Figure 1: Example of adaptive autonomic system operation in a pervasive computing environment

This example focuses on the need to integrate service adaptivity and 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.

This paper first provides some background on the types of adaptive system mechanisms we are working with and then outlines an architecture that abstracts these mechanisms. We then present in more detail some initial results that focus on the natural interaction between users and communities and the autonomic system.

## 2. Adaptive Mechanisms

We build upon the following adaptive mechanisms in considering how to integrate user-centric adaptivity and management of adaptivity constraints.

### 2.1 Service Composition

Pervasive computing environments will exhibit a large amount of heterogeneity in the components from which they are constructed. Any autonomic system supporting pervasive 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 through well defined service interfaces, allowing a pervasive 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 what 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 [21]. However, service-oriented architectures tell us little about how such adaptive services can be used to allow people to interact with a pervasive computing environment in a seamless and unobtrusive manner. Automatic service composition is usually driven by some technical specification of the overall service required, thus confounding the requirement of unobtrusive service usage for a person in a pervasive computing environment.

### 2.2 Policy-based Management

Another adaptive technique that is seeing increased deployment in network and system management is policy-based management [22]. It uses expressive rule languages to determine behavioral 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][24], 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][15]. In pervasive 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].

### 2.3 Adaptive Hypermedia

Presentation-centric adaptive systems use explicit user models to tailor information to different users. Data is collected for the user model from various sources [14], e.g. contact lists, schedules, terminal capabilities, application usage histories and security, cost, navigational and presentational preferences. The user model is the basis of the adaptation effects. One area of strong research into personalized adaptive systems is Adaptive Hypermedia systems, which are typically applied to areas of learning, such as museum guides or eLearning. These offer an alternative to the traditional “one-size-fits-all” approach by employing user models that allow personalization in hypermedia systems. The benefits of such personalization include relevancy, reduced time to learn and improved retention and recall by users. The experiences of the adaptive hypermedia community therefore provide a rich seam of techniques and architectures that may prove useful in developing user-centric adaptive services. We are assessing the usefulness of these experiences as we move from document-oriented adaptation of hypermedia to service-oriented adaptation. In general, the focus of adaptive hypermedia systems on user cognition will have to be expanded to address the adaptive delivery of services to the user. However, the conceptual similarities pointed out in [18] between the service-oriented concepts as used in architectural description languages and the hypermedia meta-model indicate that such a shift can be readily accommodated. We have already developed a sophisticated generic adaptive engine that has been applied successfully to personalized eLearning hypermedia [3]. We are currently investigating how this engine can be applied to support adaptive service composition.

### 3. Adaptive Service Architecture

Our approach to developing, operating and managing pervasive 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. 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 pervasive 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 pervasive computing environments, we have a greater need for services to autonomously adapt their behavior rather than being adapted by the action of a human developer or administrator. More specifically, services must adapt their behavior 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 behavior will include the definition of when these other services are invoked and how they are used.

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

- Is aware of context information that we assume has been made available in the pervasive 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 behavior 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 behavioral rules to the adaptive service element. These rules dictate the element’s behavior 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 behavioral rules, we are presented with two major interoperability challenges. The first occurs when adaptive systems attempt to automatically generate the behavioral rules based on the user model and the context of the task-at-hand. The second occurs when coordinating behavioral rules destined for 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 behavior the system as a whole is required to exhibit.

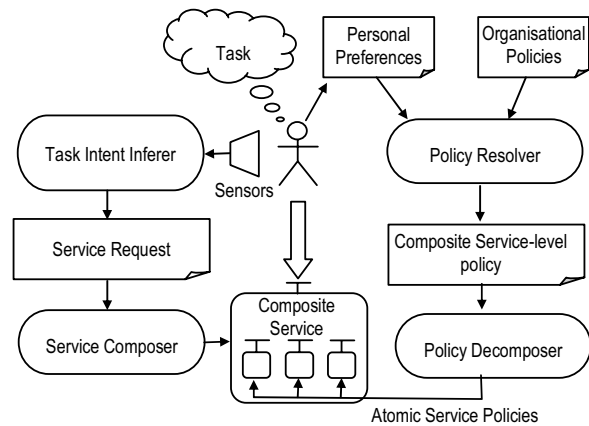


Figure 2: Integration of adaptive service composition and adaptive policy-based management

Figure 2 outlines how we expect adaptive behavior 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 behavior 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 behavioral rules, expressed both as individual personal preferences and as organizational policies, need to be resolved into behavioral 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 pervasive 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 pervasive computing environments as these will often be supporting fluid, collaborative organizational structures with distributed, overlapping responsibilities for authoring policies on resources.

To enable the adaptive system to process such behavioral semantics automatically we adopt ontology-based semantics as a means of describing constraints on an adaptive service element's behavioral rules in a machine intelligible form. The expression of behavioral 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 [20], which promise the automation of service discovery and composition [17]. Such semantic services are also highly applicable to the ad hoc, dynamic service composition needed for pervasive computing [16][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 behavioral rule vocabulary for an adaptive service element [23]. However, the issue raised by the heterogeneity of ontologies and how to achieve semantic interoperability between systems using different ontologies remains a challenge [19].

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,
- a description of the services of which it makes use,
- a rule-based model for describing and restricting its behavior.

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

Service Descriptions are specified in OWL-S language 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 behavior rules are in the form event-conditions-actions and dictate the behavior 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 behavior rules, thus restricting unwanted rule behavior. ASEs may be generated by

automated service composers and thus have entirely rule based behavior, or they may be pre-implemented software components with restricted rule-based behavior for flexibly enforcing policies to invoke external services, such as accounting or fault management, when specific events occur.

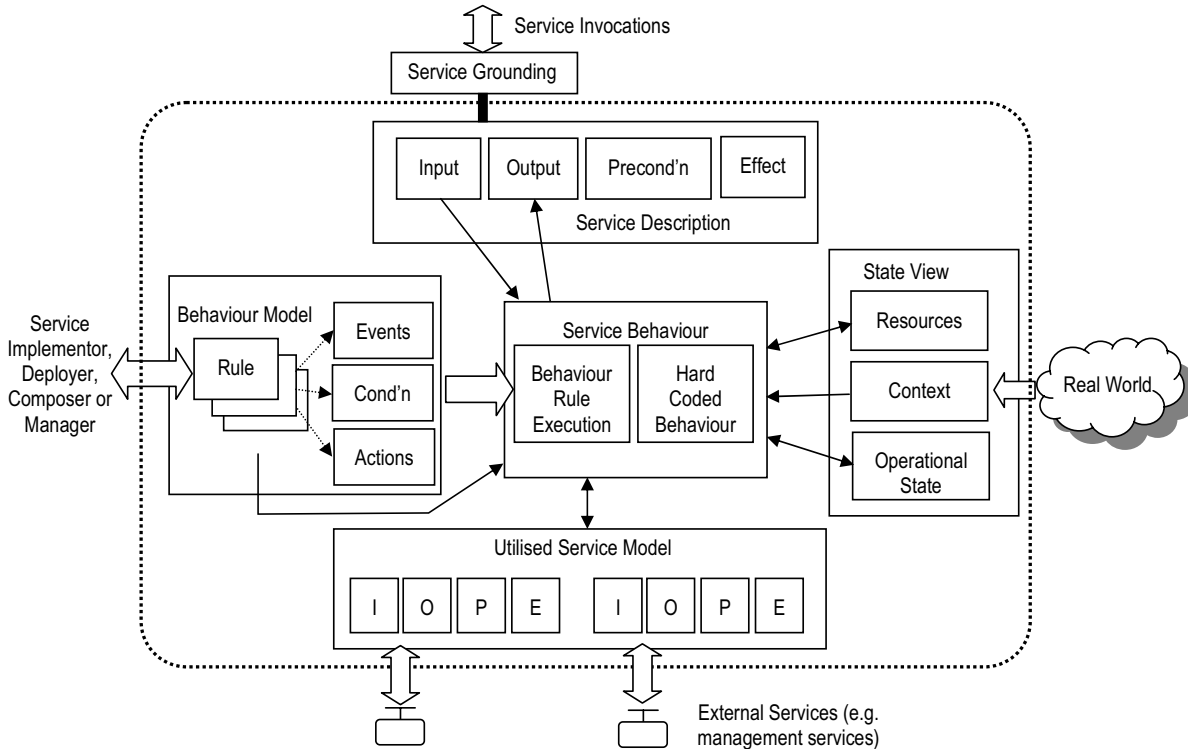


Figure 3: Anatomy of an Adaptive Service Element

The following section focuses on how we determine naturally the aims of the adaptive service user and how we can easily specify constraints on the adaptive system's behavior in complex organizational situations.

#### 4 Person Centric Adaptive Services

We have seen that research into presentation-centric adaptive systems, in particular adaptive hypermedia, have focussed on the axes of adaptation that guide how the user interacts with the system. This usually requires a fairly fixed domain of goals to select from, whereas an autonomic systems for pervasive computing will have a very open domain of individual and collective user goals that must be interpreted dynamically. Here we present initial results first in inferring individual user task goals for service adaptation and then in

resolving and de-conflicting collectively developed management policies onto resource-level policies.

#### 4.1 Inference of Individual User Intent

Given the expected proliferation of devices in a pervasive computing environment, it is unreasonable to require a user to interact directly with each of them. Similarly, a proliferation of discrete per device interfaces would make the system unwieldy in terms of user interaction. An alternative to this scenario would be to provide a single interface to the environment which would in turn handle the interaction with the resources in the space. Further user empowerment could be achieved by making this interface context aware and capable of monitoring users' natural activities as a method of gaining meaningful inputs. In essence, the pervasive computing space would be equipped with a range of monitoring devices. Some

would be passive such as video cameras and microphones. These could track motion, gestures, spoken word and prosody. In addition, we propose making use of declarative tools, which would allow less ambiguous input. Such tools might include our current keyboard and mouse combination, as well as intelligent white boards or artifacts that can be manipulated for agreed meaning, e.g. tokens within the space that are understood to represent members of a team, and boxes which represent groups to which they can be assigned; e.g. placing a token in a box assigns an individual to a group.

We are developing an interface technology, known as TSUNAMI (Tailored Support of User's Natural Activities with Mixed Initiative), that over a period of time attempts to infer a user's goals [8]. This is done by gleaning data from the sensors above, as well as using context information such as calendars, business policies and user profiles, to build a model of the user's intentions at a given time. Each of these inputs informs the overall view of the user's actions with appropriate authority. For example, the sensor data is weighted to take account of ambiguity. Similarly, evidence degrades over time so that observed data that occurred an hour ago say, does not overly inform the current analysis.

As data is built up, the system can become more confident about the user's intentions and is therefore better equipped to successfully offer pre-emptive support. Such support is offered with mixed initiative [9], in other words the point at which the system offers support depends on the task-at-hand and the preferences of the user. A utility threshold is used to govern this. The confidence the system has in its prediction is compared to the threshold required to initiate support. Once the threshold is exceeded the system assumes that the utility of offering support, even if it turns out to have been incorrect, is higher than that of not stepping in. In certain cases, for example mission critical applications, support may still require unambiguous consent from the user.

The process of associating evidence with potential support is achieved using event analysis and aggregation. As evidence is observed, it is parsed and added to previously recorded inputs to produce more complex patterns of events. These are aggregated again to produce outputs that can be dispatched as a request to a suitable service delivery mechanism, e.g. an automated service composition engine. The analysis and aggregation is achieved by means of Bayesian networks where nodes correspond to either the initial atomic inputs, their aggregates and ultimately the actual request candidates.

As an initial test of this interface system, we are carrying out a limited implementation. Our simulator offers a 2-dimensional image of an office, where input is given via mouse position tracking (to act as gaze tracking) and text input (to act as voice). Knowledge about the simulated environment, e.g. the locations and capabilities of objects within the space is stored in ontologies written in DAML and stored in a Xindice database. The mouse position is used to search the ontology to discover the device being considered.

There are three test scenarios where users will be asked to perform daily tasks such as transferring files between users, moving displayed data between terminals and making a phone call. The objective of the experiment is to prove that the basic architecture is sound, and in particular that event aggregation and analysis can be successfully achieved using Bayesian Networks.

## 4.2 Resolving Collective Constraints on Adaptive Behavior

For services and resources used solely by an individual, constraints can be specified as direct input to that person's general application preferences. The constraints can be expressed as user level behavioral rules, or policies. In multi-user settings, constraining the adaptive behavior of the services that use the managed resources is highly complex as there may be communal responsibility for deciding how resources should be shared and therefore policies need to be authored in communal fashion. The dynamic interoperability offered by pervasive computing environments means that traditional physical and organizational structures do represent intrinsic means and boundaries for the control of how mobile users can access and use resources. While this may be beneficial in terms of enabling dynamic collaboration and empowering mobile users, it also requires that the user's need for resources is dynamically and efficiently resolved against the goals of the owners or administrators of the resource.

Conventional approaches to constraining adaptive system behavior through policies typically support organization models through binding policies to roles that represent groups of users by their job function. However, the definition of roles is a centralized activity not well suited to fluid, collaborative organizational models where people have flexible, ever changing job functions. Our approach aligns the mechanism for specifying policy-rules with the natural operation and evolution of collaborating groups. The types of group structure we address range from hierarchical organizations to peer-based web

communities, but with the focus on the more acute communal policy authoring problems of the latter. Our approach adopts a community-based model for defining constraints as a more flexible alternative to role-based policy approaches. The semantics for a community-based model for the delegation of authority to different groups and sub-groups within an organization has been defined. Communities possess resources but may delegate authority to author policies related to those resources to sub-communities. This allows an organization as a whole to delegate responsibility for authoring policies to manage resources to sub-groups. An important feature of this scheme is that the detection of conflicting rules related to a policy are automatically detected at the level of community that has the responsibility over the resources concerned and which therefore has the authority to define how the conflict can be resolved between sub-communities. An initial implementation of the scheme has been developed building on PONDER, an existing policy framework [4]. This has been used to model the structure of an existing internet community with the initial aim of mirroring its decision making process and monitoring the accuracy with which the model reflects real policy conflicts and their resolution. Further details of the community-based model and its implementation can be found in [6].

## 5. Conclusions and Future Work

Overall this work addresses the need to closely integrate adaptive service composition and the policy-based constraining of adaptive system behavior. In pervasive computing environments, these mechanisms must naturally support user and organizational goals. We see the use of ontology-based semantics as a key technology in addressing these integration challenges and propose a semantic-rich, component-oriented architecture to support this integration.

We have presented initial results on how individuals can naturally specify task-related service requirements and how the specification of collective constraints on adaptive behavior can be naturally aligned to dynamic organizational structures. Overall we plan to integrate both the inference of intent and the community based constraint authoring into a framework for context-aware adaptive services. The inference of intent will be integrated with automated service composition. Several, ontology-based approaches are being considered, though we are focusing on the use of personalized composition templates using adaptive hypermedia technologies. We are exploiting semantic

representations of the resulting composite service, the constituent services and the ASE policy vocabularies for automatically mapping to policies for a composite service into policies native to constituent ASEs. In support of these activities, we are also assessing the problems in mapping heterogeneous, separately developed ontologies so that they are amenable to run-time semantic interoperability [19].

The intent inference mechanism is being extended in several ways including:

- Establish a vocabulary for service composition requests.
- Further experiments with additional user information sources, e.g. user models and schedules, location and social context, as well as provision of some information sources via real sensors.
- Support for authoring Bayesian Networks for particular pervasive computing environments.

For the community-based constraint authoring we plan to address:

- Deployment of the community-based policy system for the management of CVS repositories, feature requests and bug reports for a live, Internet-based software development community. This will provide an initial assessment of the acceptability of community-based policy management.
- Assessing the impact federation between communities on community based policy management.
- Integrate constraints based on service/resource semantics to aid identification of conflicts in policy authoring.
- Develop user interfaces for community management, policy authoring and conflict management and deploy in experiments with live internet communities.

## Acknowledgements

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

## 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.
- [3] Conlan, O., Wade, V., Bruen, C., Gargan, M. (2002), "Multi-Model, Metadata Driven Approach to Adaptive Hypermedia Services for Personalized eLearning", Proceedings of Second Conference on Adaptive Hypermedia and Adaptive Web-Based Systems, Springer, LNCS 2347, May 2002, pp 100-111
- [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" Proc of DAIS 2003, Lecture Notes in Computer Science 2893, pp. 266-273, International Federation for Information Processing, 2003
- [9] Horvitz,, "Principles of Mixed-Initiative User Interfaces", Proceedings of the SIGCHI conference on Human factors in computing systems, pages 159-166, 1999
- [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] Kobsa, A., "User modeling: Recent Work, Prospects and Hazards", In Adaptive User Interfaces: Principles and Practice, 1993, North-Holland: Amsterdam.
- [15] 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.
- [16] Masuoka, R., Labrou, Y., Parsia, B., Sirin, E., "Ontology-Enables Pervasive Computing Applications", IEEE Intelligent Systems, Sept/Oct 2003, pp 68-72
- [17] McIlraith, S.A., Son, T.C., Honglei Zeng, H., "Semantic Web Services", IEEE Intelligent Systems, 16(2), March/April 2001
- [18] Muchaluat-Saade, D.C., Soares, L.F.G., "Towards the Convergence between Hypermedia Authoring Languages and Architecture Description Languages", Proceeding of the ACM Symposium on Document Engineering, Atlanta, Georgia, USA, ACM Press, pp 48-57, 2001
- [19] 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
- [20] "OWL-S: Semantic Markup for Web Services", The DAML Service Coalition, <http://www.daml.org/services/>, October 2002
- [21] 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
- [22] Sloman, M., Lupu, E. "Security and Management Policy Specification", IEEE Network, Vol. 16, no. 2, 2002, p 10-19.
- [23] Tonti, G., Bradshaw, J.M., Jeffers, R., Montanari, R., Suri, 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
- [24] 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, Deconflicts, and Enforcement", IEEE 4th International Workshop on Policies for Distributed Systems and Networks, June 04 - 06, 2003, Lake Como, Italy, pp 93-98
- [25] Weiser, M., "The Computer of the 21st Century", Scientific American, vol. 265, no.3, September 1991, pp 66-75.