Person Centric Service Adaptation

Dave Lewis, Tony O'Donnell, Kevin Feeney, Aoife Brady
Knowledge and Data Engineering Group
Department of Computer Science
Trinity College Dublin

Abstract

Adaptive systems can be modelled as being constructed from Adaptive Service Elements that, individually and collectively, use resources to provide users with services and then adapt how those services are delivered based on changes in the system's context and available resources. People interact with such adaptive systems both as users of services and as those responsible for the allocation and use of the resources employed, e.g. network bandwidth, file access, device displays, server cycles etc. Person Centric Service Adaptation addresses how the needs of such users are interpreted and mapped onto the adaptive mechanisms of the adaptive system and its components. Adaptation must be conducted in a manner that is both natural to use and transparent to the user, while at the same time ensuring the user has a sense of control over the adaptive mechanisms.

In this paper we review how adaptive systems have largely focussed on how a user interacts with an adaptive system. We argue, however, that for adaptive systems to be deployed in ubiquitous computing environments we must address more natural interaction with the system, enabled by the wider range of multi-model devices available. We also argue that for wide-spread deployment of adaptive systems, stronger abstractions must be developed in the constraining of adaptive behaviour, especially when adaptive ubiquitous computing systems are used in organisations and collaborative activities. Initial work towards these goals is presented.

1. Introduction

Adaptive systems are ones which adapt their behaviour to changes in context and resources. Ubiquitous 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. We can therefore consider a ubiquitous computing environment that is aware of a user task context and the resources it has to support user tasks as an adaptive system. We envisage, however, that ubiquitous computing environment will exhibit a large amount of heterogeneity in the components from which it is constructed. Any adaptive system supporting ubiquitous computing will therefore face major interoperability and integration challenges. These can be eased somewhat by adopting a service-oriented architecture, where constituent elements are access via a well-defined, self-describing interface. In adopting such an architecture we therefore need to think in term of adaptive service elements being used to dynamically generate new adaptive service to satisfy user task requirements. Research into adaptive services tends, however, to focus on the mechanisms for adaptation, e.g. service composition of behavioural rules, rather than focussing on modes of adaptation that empower the user to do want they want in the manner they want to do it. In this paper we examine some of the existing mechanisms for adaptive services that we see as useful for ubiquitous 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 behaviour they require from an adaptive system. 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 organisations or communities. We do not focus in this paper on how the adaptive service interacts with the user or on the mechanisms of the adaptivity. Instead we focus on how the people, 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 behaviour they require at any particular point in time?
- How the user can exert their responsibility over any resources that may be adaptively allocated to service usage?

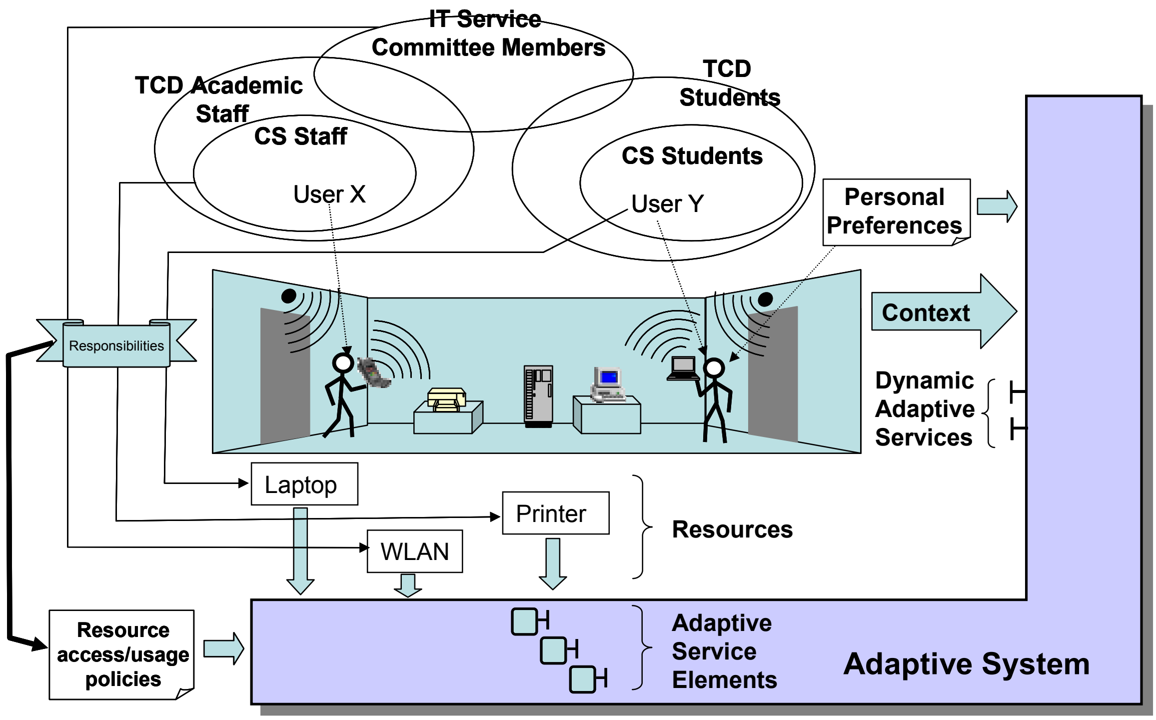

Figure 1: Example of adaptive system usage in a ubiquitous computing environment

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 they can maintain a sense of ownership over adaptive service behaviour.

Consider the scenario of a university student coming to meet a lecturer for a tutorial session. 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 services and desktop terminals. Individual resources are made available through adaptive service elements which are used by the adaptive system which implements the ubiquitous computing environment, to generate the services needed by the student and the lecturer, e.g. one 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 service that should be offered to the users at any point in time. However, this adaptive process should also be informed by default behaviour contained in personal preferences as well as restrictions on what the people involved are permitted to do, e.g. can 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 universities organization, including, for instance, information service committees which set wireless LAN usage policies. Such a scenario points to the need for the adaptive system to mediate between, sometimes imprecise user intent and complex organisation policies and the adaptive mechanisms that generate and manage the tailored services provided to the user.

We first provide some background on the types of adaptive system mechanisms we are working with and outline an architecture that abstracts these mechanisms. We then present initial result in two approaches.

2. Background

2.1. Mechanisms for Adaptive Systems

Service-oriented system architectures provide a way of flexibly combining functional elements in different ways. 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 []. 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 they wish to perform it in. 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. Automatic service composition is usually driven by some technical specification of the overall service required, thus confounding the requirement for unobtrusive service usage for a person in a ubiquitous computing environment.

Another adaptive technique that is seeing increased deployment in network and system management is policy-based management. It uses expressive rule languages to determine behavioural rules for how a system should respond to predetermined event and system conditions. Though policy-languages have been developed that can express policies at a relatively high level of abstraction [][], automatically mapping these to rules that can operate on heterogeneous, low level resources is problematic. 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 trying to be achieved and the semantics of the resources used to achieve those role. In ubiquitous computing environments, anyone entering the space may possess or use resources that may be shared and therefore potentially managed via policies. Existing policy based languages are not amenable to such widespread application amongst the use community.

User-centric adaptive systems use explicit user models to tailor information to different users. Data is collected for the user model from various sources. 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 area of learning, such as museum guide or eLearning. These offer an alternative to the traditional "one-size-fits-all" approach by the use of user models to allow personalisation in hypermedia systems. The benefits of such personalization include relevancy, reduces time to learn and improves retention and recall.

Brusilovsky defines four high level classifications that define adaptation effects []. The four axes of adaptivity are as follows:

- Adaptive Navigation: attempts to guide the user through the system by customising the link structure or format according to a user model.
- Structural Adaptation: attempts to give the user a spatial representation of the Hyperspace environment, based on the user model, aiming to give the user an idea of where they are within the system.
- Historical Adaptation: attempts to give a time context to the user by adapting representations of the user's path through the system.
- Adaptive Presentation: is the customisation of the content to match user characteristics specified by the user model.

The experiences of the adaptive hypermedia community provide a rich seam of techniques and architectures that may prove useful in adaptive services. One challenge is assessing the usefulness of these experiences as we move from document-oriented adaptation of AH 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 [] between the service-oriented concepts as used in architectural description languages and hypermedia 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. An accompanying paper discusses the issues in applying this platform to adaptive services for ubiquitous computing environments.

2.2. Adaptive Service Architecture

This work is related to an evolving, service oriented, abstract model of adaptive systems. It is based on the assumption that all functionality in a ubiquitous computing environment this 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 other service implementations. Ideally, services should represent the only way in which these resources can be manipulated via a computing system, though backward compatibility issues may 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 service that use SOAP, WSDL and UDDI infrastructure. However, when applied to ubiquitous computing environments, we have a greater need for these 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 how and when these other services are used.

We model a service and its behaviour using the abstract concept of an adaptive service element. 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 adaptive service elements 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 adaptive service element may need to be managed to reflect the needs and preference 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 elements' behaviour within the constraints provided by the elements' developers, be they human designers or automated agents that perform service composition.

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 destined for adaptive service element from different sources. In both cases mappings need to be established between the semantics of the rule constraints of different adaptive service elements and the semantics of the behaviour the system as a whole is required to exhibit. To enable adaptive systems to process such semantics automatically we are adopting 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 these constraint semantics is eased by having the semantics of services and the operational context also expressed in an ontological format. Ontologies for service specifications are already emerging under the semantic web community [], while the use of the ontologies for expressing context information has also recently been explored [][][]. Such ontologies will provide the basic terms that will make up the behavioural rule vocabulary from which the permissible rules for an adaptive service element can be formed. An accompanying paper outlines some of the issues raised by the heterogeneity of ontologies and how to achieve semantic interoperability between systems using different ontologies []. The following section focuses on how we determine naturally the aims of the adaptive service user and how we can easily specify constraints on adaptive system behaviour in complex organisational situations.

3. Person Centric Adaptive Services

We have seen that research into user-centric adaptive systems, in particular adaptive hypermedia, have focussed on the axes of adaptation that guide how the user interacts with the system. This has usually required a fairly fix domain of goal to select from, whereas adaptive system for ubiquitous computing will have a very open domain of user goals which must be interpreted dynamically. In addition, the use of adaptive systems in a ubiquitous computing environment will involve those systems being used collaboratively by user and also being subject to goals of user in different role. For the latter, we must focus, in particular on resolving the goals of users with the goals of those responsible for resources. The dynamic interoperability offered by ubiquitous computing environments means that traditional physical and organisational structures do represent intrinsic mean of controlling how mobile users can access 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.

The following section report on initial results from two threads of research conducted un the M-Zones project

3.1. Understanding what the user wants to do

Given the expected proliferation of devices in a ubiquitous 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. 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 ubiquitous 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 artefacts that can be manipulated for agreed meaning.

Over a period of time the interface technology, known as TSUNAMI (Tailored Support of User's Natural Activities with Mixed Initiative), attempts to infer a user's goals [higel et al, 2003]. 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[Horvitz?], 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. 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 propose carrying out a limited implementation. In this 2-dimensional simulation of an office, input will be given via mouse position tracking (to act as gaze tracking) and text input (to act as voice). There will be 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 phonecall.

3.2. Specifying Constraints on Adaptive Behaviour

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 behavioural rules, often known as policies. For shared resources, constraining the adaptive behaviour of service use to access them is highly complex as there may be communal responsibility for deciding how resources should be shared and therefore polices need to be authored in communal fashion. There is some experience in applying policies to both define and constrain adaptive behaviour in large organisation, especially for the purposes of system access control and network management. However none of these approach, or more advance policy frameworks form the research community, address in any depth the problem of communal policy authoring. We aim to align the mechanism for specifying the policy-rules with the natural operation and evolution of collaborating communities, from hierarchical organisations to peer-based web communities, but with the focus on the more acute communal policy authoring problems of the latter.

The work conducted in this area proposes a community-based approach to defining constraints as a more flexible alternative to current role-based policy approaches. Semantics for a community based model for the delegation of authority to different groups and sub-group within an organisation has been defined. This allows an organisation 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 natural responsibility over the resources and which therefore has the authority to define how the conflict can be resolved. An initial implementation of the scheme has been developed building on an existing policy framework called PONDER. 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 reflect real policy conflicts and their resolution.

4. Future Work

Overall we plan to integrate both the inference of intent and the community based constraint authoring into a framework for context-aware adaptive services. This framework makes heavy use of ontology-based semantics, which we will continue to integrate into both these approaches to model the resources and adaptive services with which they must integrate.

Related to the intent inference mechanism, this will be extended in several way including:

- Establish vocabulary for interacting with adaptive service mechanism as basis for integration
- Further experiments with additional user information sources, e.g. use models and schedules, location and social context, as well as providing some information sources via real sensors
- Introduce user feedback and learning mechanism into inference loop

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

- The impact on the model of community federation
- 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 experiment with live internet communities.

5. Conclusions

We are examining a number of techniques for making the use of adaptive services mechanisms more natural for users and communities, in particular the automate inference of user intent from natural user interaction and the natural integration of adaptive behaviour constrain engineering into organisational decision making processes. Through these techniques we intent to empower the user both as individuals and in communities by allowing them to govern adaptive behaviour of ubiquitous computing environment in as natural a manner as possible.

Acknowledgements

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

References

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

[horvitz, 1999] Principles of Mixed-Initiative User Interfaces, Horvitz, Eric, Proceedings of the SIGCHI conference on Human factors in computing systems, pages 159-166, 1999