Next Generation Context Aware Adaptive Services

Owen Conlan, Ruaidhrí Power,
Steffen Higel, Declan O'Sullivan
Knowledge and Data Engineering Group (KDEG)
Trinity College, Dublin
+353 1 6082158
{oconlan, rspower, higels, osulldps}@cs.tcd.ie

Keara Barrett
Telecommunications Software and Systems Group (TSSG)
Waterford Institute of Technology
+353 51 302900
kbarrett@tssg.org

ABSTRACT

Situational information can enrich the interactions between a user and the services they wish to utilize. Such information encompasses details about the user, the physical environment and the computing resources. There are at least three key aspects in addressing this issue. Firstly, it is important to accurately capture or infer the requirements of the users in a timely fashion. Without precise information on what the users are hoping to achieve it is difficult to identify suitable services or sub-services that may fulfill (in part or fully) their information needs. This information may be obtained by combining information inferred about the users with that entered through direct user control mechanisms, thus empowering the users. Secondly, the nature of the available services determines the modes in which they may be adapted to the users' needs. Rigid, inflexible services may be difficult to tune to the information requirements of the users. Adaptive services, on the other hand, are well suited to dynamically modifying their behavior, within defined constraints. The third issue to be addressed is the on-the-fly combination of services to meet the users' requirements. A single service attempting to fulfill all of the users' requirements may suffer from a bloating problem. Adaptive services, coupled with information about the users, are central to the ability to assemble ad-hoc conglomerate services from sub-services. The service delivering information to the user must, however, be able to utilize this information effectively to contextualize and adapt its output to the user. Adaptive services should be able to react to the user's surroundings as well. By combining adaptive services the potential for utilizing context information is increased. Finally, an adaptive service should be conscious of the computing context in which it is operating. This paper argues that current modeling (both of users and services) techniques, adaptive axes and personalization techniques used in current personalized information services, such as Adaptive Hypermedia Systems, may supply the basis for next generation adaptive collaborative services.

Categories and Subject Descriptors

H.1.2 [**User/Machine Systems**]: Human Factors

Permission to make digital or hard copies of all or part of this work for personal or classroom use is granted without fee provided that copies are not made or distributed for profit or commercial advantage and that copies bear this notice and the full citation on the first page. To copy otherwise, or republish, to post on servers or to redistribute to lists, requires prior specific permission and/or a fee.
Conference '00, Month 1-2, 2000, City, State.
Copyright 2000 ACM 1-58113-000-0/00/0000…$5.00.

General Terms

Experimentation, Theory.

Keywords

Adaptivity, Modeling, Context, Services.

1. INTRODUCTION

Current services supporting users often fail to meet the rapidly changing requirements of users as their information requirements are often focused and may be short lived. Collaborative work activities present a challenging *personalization* situation – no longer are we adapting to an individual in a well understood domain, as in personalized eLearning. Rather we are adapting to a group of users involved in a collaborative process. This process, or the desired outcome, may not be as well defined as in the case of a personalized eLearning service. There are, however, a number of potential sources that may be used to acquire this information. The primary of which are the users themselves, or more accurately the context information stored about the users. A group of users will bring a wealth of information to any task they are performing. This information combined with some information about the business outcomes they are hoping to achieve form the knowledge basis of the service adaptation and combination. An understanding of the types of information requirements and possible uses of that information are the secondary source of data.

Adaptive services base their adaptivity on user and context information, as well as on an encapsulation of the expertise that support the adaptation. In the case of an eLearning service the personalization of educational material may be based on the learner's learning goals, prior knowledge and learning preferences, i.e. user context information. It is also supported by the expertise of an expert in the knowledge domain and an instructional designer, who has the appropriate pedagogical expertise. Adaptive services should, therefore, be able to accept, and understand, input to their adaptive processes from many sources.

In the case of an adaptive service fulfilling a business requirement, the sources may include context information about the users, information about the environment(s) in which they wish to collaborate, restrictions of the devices they are using and a model of business process, to name a few. These sources of information are not static – users change their goals, sound and lighting aspects of environments change, user may use multiple devices etc. It is unlikely, given the large variance of information sources, that a single adaptive service will be able to fulfill anything more than a few basic information needs of a user.

It is more likely that the fulfillment of such varied requirements will stem from the combination of a number of services. By combining the facilities offered by a number of services in an ad-hoc fashion the information needs of a user or group of users may be met. The services must, however, be combined in a meaningful way, be adaptive to the changing requirements of the users, adapt to the changes of external factors (environment, device etc.) and be fault tolerant.

The three aspects of capturing users' requirements, designing adaptive services and combining those services in a meaningful way present a number of research challenges in the areas of –

- **User Modeling –** acquiring the requirements of the users and reconciling this with their historical information
- **Context Modeling –** appropriate context information concerning the users, environment and required business outcomes of the information personalization is available
- **Semantic Interoperability** – ensuring that the information about users and business outcomes is compatible
- **Multi-modal Information Presentation** – the same content may potentially be rendered on many different devices and should be adapted to the characteristics of a device
- **Service Composition** – the composition of new services on-the-fly from existing services or parts of services
- **Self Management of Services** – with the potential for changing context information and service failure the services should be capable of adapting to these changes in situation

This paper discusses the research challenges (Section 2) associated with next generation adaptive services, focusing on context information (Sections 3, 4 and 5) and how it can be used to fuel adaptive services (Section 5) and their composition (Sections 6 and 7). The case study of services for personalized eLearning (Section 8) is used to illustrate many of these principals in the given domain.

2. RESEARCH CHALLENGES

This section will briefly discuss some of the research challenges faced in trying to realize the view of dynamically adaptive collaborative services.

2.1 User Modeling

The capturing of appropriate user model information is a common issue in personalization systems. What is less common is the reconciliation of multiple user models to form the basis of adaptation. A user model contains explicitly modeled assumptions that represent the characteristics of the user which are pertinent to the system [20]. The system can consult the user model to adapt the performance of the system to each user's characteristics. This view is changed somewhat considering that we are no longer concerned with a single user. The system, also, may not be a single entity, rather a combination of several services. There are several techniques for modeling users and refining this model –

- **Stereotype Model** – creating fixed stereotypes is one of the simplest ways of user modeling [27] [23].
- **Overlay Model** – A model of the user knowledge is constructed on a concept-by-concept basis and updated as the user progresses through the system. This allows for a flexible model of the user [7].
- **Combination Model** – the Stereotype and Overlay techniques of user modeling are often combined in educational adaptive hypermedia systems. The user may be categorized by stereotype initially and then this model is gradually modified as the overlay model is built from information acquired from the user's interaction with the system.

There are a number of implicit approaches that may be used in acquiring and refining the user model. These include –

- The observation of the user's direct-manipulative interaction with the software system.
- The analysis of the information which the user retrieves from a database or repository [22].
- The system can also explicitly ask the user for information [23] employing mechanisms such as questionnaires.

The reconciliation of the user information with other models in the system impacts upon the context information stored about those models and how interoperable the information in those models is.

2.2 Context Modeling and Semantic Interoperability

Context refers to information that relates to a situation. This typically includes information about the user, environment and potentially the task they wish to achieve. It is important that this information can be gathered (see *User Modeling* above) and shared. The sharing of such information also presents research challenges. Traditionally when interconnecting information systems of different parties, design time solutions (e.g. handcrafting of gateways) have been used to bridge semantic differences that may exist in the information of both parties. Increasingly due to the rapidly changing nature of relationships between parties (e.g. in the B2B area), a solution that allows for semantic interoperability to be achieved at runtime through the dynamic brokering of meaning and the dynamic translation of dialogue between parties is needed. This need for semantic interoperability or shared understanding is particularly visible in the ad-hoc combination of services to fulfill the requirements of a group of users.

2.3 Multi-modal Information Presentation

Multi-modal information presentation involves adapting the material delivered to a device to the characteristics of that specific device. For example, PDAs typically have much less screen real estate than desktop PCs. In this case it may be necessary to modify the content to ensure it renders correctly on the target device. In broad environmental situations users may be using a heterogeneous set of devices to view the same content over time. In these situations equivalent renderings of the content should retain the key features of interest to the users to facilitate effective discussion of that material.

2.4 Service Composition and Self Management of Services

Service Composition is the orchestration of a number of existing services to provide a richer composite service assembled to meet some user requirements. The current major interest in service composition, however, stems from the emergence of web services and the possibility of composing them to provide value-added services over the WWW. Service composition techniques typically involve expressing elemental services and composite services, the latter being compositions of elemental services and other composite services. The definition of composite services requires the expression of the flow of control and information between the elemental services. One of the challenges of combining services in an ad-hoc fashion is that of reliability.

3. DEFINING CONTEXT

Much debate has occurred and is still taking place about the meaning of both context and context-aware computing. Therefore one of the first steps in producing a context management system is to determine what information constitutes context.

Schilit and Theimer, the pioneers of context-aware computing, regard context to be location, identities of nearby people and objects, and changes to those objects. They consider where you are, whom you are with, and what resources are nearby to be the important aspects of context. Abowd et. al.'s more recent classification of context [1] expands the Schilit et al definition. They define context as:

"…any information that can be used to characterize the situation of an entity. An entity is a person, place or object that is considered relevant to the interaction between a user and an application, including the user and application themselves."

This means that any information that depicts the situation of a user can be entitled context. The temperature, the presence of another person, the nearby devices, the devices a user has at hand and the orientation of the user are examples.

When humans talk directly to one another they are able to use implicit information about the situation, i.e. context, to enhance the conversational bandwidth. Unfortunately this implicit information does not transfer naturally to human-computer dialogue [17]. The concept behind context-aware computing is to exploit the progress in sensing and mechanisms for observing the environment to systematically collect implicit context [3].

Context information can be formed into an abstract model of all the actors in a smart space system. A system is context-aware if it uses the context information in the abstract model to provide relevant information and/or services to the user. Context-aware systems can make more informed decisions about information to be presented to users and how to react to commands received from users. According to Schilit and Theimer context-aware computing is any system that "adapts according to its location of use, the collection of nearby people and objects, as well as changes to those over time" [29]

Context is effective only when shared [34]. To ensure context is shared, context must first be gathered and managed by a context-aware system. This implies that the context-aware system must understand what context is before it can go about seeking and categorizing this information. Schilit et. al. propose the following classification of context information: [29]

- **Computing Context** - network connectivity, bandwidth, and nearby resources such as printers, displays, or workstations.

- **User Context** - the user’s profile, location, nearby people, and current social situation.

- **Physical Context** - lighting, noise level, traffic conditions, temperature.

Each of these categories contains a wealth of information relevant to the context-aware system. They cannot however be used in isolation to full effect; computing context will be combined with user context, and user context will be combined with physical context to provide a full picture for the context-aware system.

3.1 Physical and Virtual Context

Context as defined in section 3 may alternatively be subdivided into two categories: Physical Context and Virtual Context. The intention of a context-aware system is to gather both physical and virtual context and merge them together to achieve an overall picture of the situation.

Virtual Context may include the version of the operating system, the interface capabilities, the wireless technology used to accomplish communication, email messages sent and received, and documents edited.

Physical context on the other hand may be the presence of another entity, be it a user or device, the proximity to a particular printer, information indicating if the user is standing, walking or sitting or the current weather conditions.

When both physical and virtual context are unified a different reasoning about the user’s situation may emerge. In the past the main focus was on physical context, with a particular emphasis on location, however there is now a move toward a vast range of possibilities. After both the physical and virtual context is gathered and stored in a buffer or repository, a proficient system would merge them and decide what is relevant to the user at the present moment.

3.2 Context history

Historical context information is where computing, user and physical contexts are stored across a time span. Potential uses for this stored information would be to establish patterns of smart space usage. Historical context information is particularly useful for mobile-aware applications. For example applications that predict resource consumption based on observed patterns of mobility and usage. Su et. al. describe a method of mobility prediction in [31].

Historical context is generally considered to be useful, but it is rarely used in current context-aware systems. Firstly, the context-aware system must decide what historical information is worthy of being kept, and at what level of precision. Evaluating all historical context information that is collected, to acquire the necessary information would be prohibitively costly, and efficient algorithms must be implemented to process the wealth of information available and extract meaningful data.

4. GATHERING CONTEXT DATA

Some context information can be given to the system explicitly, such as a user’s name or age; other context information can be obtained through the use of sensors.

Push and pull are the two options available to a context-aware system to extract the necessary context information from context sources. Context information is periodically sent to the service in the push model. This means that the context source collects the context information before it is needed, which may result in a better performance. The shortcoming of this approach is the consumption of resources for gathering and disseminating context that may never be exploited by the context service.

The pull model in contrast gathers only the context information that is required by the service. However, this approach exposes the context service to network delays and unavailability.

In the case of the push model, a trade off between freshness of context information with the cost of frequent updates must be measured [19], while the pull approach must consider the trade off between reserving resources and the probability of faults.

There are many types of sensor already commonly in use. These can provide primitive physical information such as light, heat and pressure readings.

Other types of context, such as facial recognition, rely on fairly simple sensors such as cameras, but require considerable processing such as image recognition in order to make use of the information obtained.

Location is an important element of context information. Many different approaches have been taken to determine the location of agents within a context-aware system. GPS, Infrared and radio signals have all been explored. Many context-aware systems that have been produced are only aware of location information. While these applications are useful, location is only one element of the wider context information. Schmidt et al discuss this fact in [30].

Sensors are not always 100% accurate or reliable, particularly if they are disposable. The information gathering system must be tolerant of sensor failure, and any information gathered from sensors must be subjected to sanity checks to help verify its correctness. Sensor fusion is one method of avoiding this difficulty.

Sensor fusion means aggregating the results of different sensors together to produce a reasonable approximation of the state of the system. This means that some sensors can fail or give erroneous answers, but the system will still be able to determine the real state through the use of a voting mechanism. When considering the output of temperature sensors for example, it might be prudent either to simply average the results or else to discard reported values that differ too greatly from what other sensors report. This method would avoid drastic measures being taken by the context system to correct what it considers to be temperature variations but are actually simply the result of sensor failure.

Sensor discovery is another issue. With the proliferation of sensors that will be required for a context system to be useful, configuring and managing a large number of sensors can be a difficult task. It is possible (perhaps even likely) that there would be more sensors than people in such a system, so being able to automate the detection and configuration of sensors would be ideal. The MUSE project [9] uses Jini for automatic sensor detection.

Location and identity are the most frequently-sensed pieces of context. Active Badges [32] produced by Olivetti and AT&T emit infrared signals which give a rough location and ID. Optical systems for context determination are also possible and research is underway in the areas of object recognition, optical tracking and motion detection, and stereo and 3D reconstruction.

The user's location, for example, would be valuable when delivering a weather report. If the user is receiving the weather information over a low-bandwidth connection it may be inappropriate to deliver graphically rich weather maps. The weather service may consult the user's calendar service to discover their scheduled location for the next day in order to deliver the appropriate weather forecast.

After sensing, acquiring and saving the context from various sources the next activity for the context-aware system is to produce a mechanism for achieving context interpretation, so that the gathered context information is utilized in a fitting manner. "Interpretation refers to the process of raising the level of abstraction of a piece of context." [16] The interpretation may involve integrating numerous contexts into one to provide a higher-level context, thus the interpreter alters context information by raising its level of abstraction.

One approach is to use context fusion to convert the "lower level context into higher level usable by applications". [24] Context fusion may be viewed in different ways. One could regard it to be the synthesis of context from the same type of sources. This synthesis increases the validity of the information so that erroneous sensors or readings are detected in order to avoid improper decisions by the system. Context fusion is also deemed to be the aggregation of context of varying types from a different variety of sources to produce a context that is exploitable by the system.

5. CONTEXT AWARE ADAPTIVE SERVICES

Chen and Kotz [12] divide the awareness that applications have of context information into two different types –

- **Active Context Awareness** - an application automatically adapts to discovered context, by changing its behavior.

- **Passive Context Awareness** - an application presents the new or updated context to an interested user or the application makes the context persistent for the user to retrieve later.

Once context information has been obtained, the use of passive context information is quite easily implemented in an application. Simply displaying the data from a sensor or the current location of a person would be uses of passive context.

The case of active awareness of context is the more difficult as it requires fundamental changes in the way traditional applications work. While the user is very much 'in control' of the current generation of applications (requiring the user to initiate almost all action), there is the danger that users could start to feel helpless and out of control of systems and services that begin to make decisions on their own because they have more information than the user does. This is a key lesson that was learned in the field of intelligent tutoring systems (ITS). After the ITS had gathered the information required (user context) it produced and delivered a personalized course to the learner. The learner, however, was rarely given the ability to adjust the delivered course or ask the ITS to rebuild the personalized offering based on new information.

With context-aware systems, services may exist that can adapt and need less explicit input from the user. For example, many devices could use context information to perform tasks such as turning on and off depending on whether they are (or are likely to be) used, using a discreet mode depending on situation, sharing a controlled amount of information with other services with the user's authorization, adapting information output based on terminal types, etc.

For services to provide support for these capabilities it is imperative that the behavior of their features is adaptable. In this way a service may offer the user only select features depending on the circumstances. The user's attention, therefore, is not cluttered by extraneous features that they do not require to fulfill their goals. If the services make good decisions, based on the information available to them, the user will perceive a better result and be more productive.

Conversely a single service may not have the features required to fully facilitate the user in their task. In this situation it would be desirable to be able to combine services, or individual features of services, into a new service to fulfill the user's requirements. This ad-hoc composition of services presents a number of research challenges.

6. ADAPTIVE SERVICE COMPOSITION

Service Composition is the orchestration of a number of existing services to provide a richer composite service, assembled to meet some user requirements. The current major interest in service composition, however, stems from the emergence of web services and the possibility of composing them to provide value-added services over the WWW. Service composition techniques typically involve expressing elemental services and composite services, the latter being compositions of elemental services and other composite services. The definition of composite services requires the expression of the flow of control and information between the elemental services. Techniques for this draw heavily on business process modeling and languages for enactable work flows. Service composition also overlaps with software engineering in the assembly of systems from pre-existing software components. Architectural Description Languages (ADL) address system assembly by assuming components offer well-defined services which are composed to meet system requirements. ADLs address static aspects of such composition, including the use of connectors to express the positioning of protocol or data transformation functions between services.

In context aware environments users (or more likely their agents) will be faced with a changing array of local services, plus varying access to remote web-based services, as users move between environments. The task of orchestrating these services to meet the needs of whatever tasks the user currently wishes to undertake therefore requires adaptive service composition, i.e. the rapid composition and re-composition of services. The problem of how to dynamically compose context aware services with little a priori knowledge is therefore analogous to the problem of web service composition.

Chakraborty and Joshi [11] define a difference between *proactive* and *reactive* composition of services. Proactive composition is performed off-line for deployment on stable, always-up, resource rich platforms. Reactive composition assembles compound services, created on the fly under the auspices of some composition manager, often optimizing for real-time parameters, e.g. available network bandwidth. With the introduction of context into this situation one can envisage a combination of reactive and proactive techniques being utilized, i.e. where possible a compound service will be assembled proactively leaving *holes* to be filled reactively later. This approach is analogous to the candidacy/abstraction model implemented in Conlan et al [15], where content holes in a personalized eLearning offering are filled at real-time with the most appropriate candidate. The candidate selection is based on context information about the learner. Real-time selection of appropriate candidates allows the service to utilize the most current information. This approach conforms to the *optional-composite* [11] view of services where not all components need to be in place for service operation.

7. SERVICE DISCOVERY

Service discovery is the process of locating what services are available to take part in a service composition. The methods used for service discovery vary greatly depending on the environment in which the services exist. For a large scale local area network, a single database of online services might be appropriate. Service discovery is completed in this manner by the DySCo system [26], which uses a single service description manager, which keeps a database of all available services and the role they can play in a compound service. For a huge array of services existing on a network on the scale of the Internet, a hierarchical lookup system like the domain name system might be the best way to achieve scalability. The eFlow system developed by HPL addresses service discovery in this manner [10]. If a service to complete the selected task cannot be found, the local service broker will ask any known external brokers if they know of the required service.

For an ad-hoc wireless network, such as those envisaged for many context aware environments, the problem becomes more complex and a further set of considerations must be made. The Anamika architecture [11], which focuses on service composition in an ad-hoc environment, uses the service discovery features of Bluetooth to locate devices within range of the node requesting a service. Each device which replies is then instructed to forward the request on to any devices they have within their range if the service is not located. This process is repeated recursively until the desired service is found.

8. CASE STUDY – ADAPTIVE SERVICES FOR eLEARNING

Course Management Systems (CMS), such as Blackboard [6] and WebCT [33] in the eLearning domain are typically monolithic systems with a fixed (i.e. non-extensible) set of features. This has led to individual CMS vendors attempting to differentiate their products from their competition by promoting and developing some features of their product. The downside of this approach is that certain features of the product may be weak by comparison. For example, a CMS may have strong learner management facilities, good content management, but have poor learner collaboration tools.

This problem may be tackled by separating the individual features into discrete services, allowing the customer to assemble a conglomerate CMS with the features they desire. There are, therefore, some complimentary research themes between this approach in the eLearning domain and the challenge of dynamic assembly of personalized services for business. The key element missing from the current eLearning research is the dynamic assembly of the services. This issue, however, is analogous to the inner workings of adaptive hypermedia systems and may be resolved by applying some of the research themes addressed in personalized content assembly and delivery to automated service combination.

The research goal of our group was to develop a framework for dynamically composing eLearning services that combine the attractive features of the modern re-usability approaches to eLearning with the power of adaptive hypermedia systems based on the open standards of bodies such as ADL [4], AICC [5] and IMS [21]. It was clear for us that the target framework should keep the winning features of the re-usability approach allowing teachers to structure a course according to their specific needs while also helping them to re-use existing relevant learning content instead of creating everything from scratch. At the same time, we want to enable the teachers to create adaptive courses and re-use not just files, but any interactive and collaborative learning activities. The framework in which the courses are delivered should also support the addition of auxiliary services, such as learner context services, collaboration tools and assessment services.

The first key feature of our solution, implemented as part of the EASEL IST Project [18], was to separate the course management system from the content. In our vision, the course management system is not a storage for all educational content to be delivered (as it is today), but a portal that provides a structured access to educational content without storing it. The content itself comes directly from different content services that are independent from any portal and generally reside on different servers distributed over multiple locations. Portals are maintained by course providers while content services are maintained by content providers. Many portals can use the same content service in very different circumstances.

The second key feature was to separate content specification from the real content. In the current model the search for the relevant content starts with some kind of content specification in terms of duration, pedagogical type, covered topics, etc. The teachers then attempt to find the desired content in a repository by issuing a formal search query in terms of content metadata. Finally, the relevant content is manually selected, copied, and integrated into the course. In our model, the teacher is able to stop at the stage of desired material specification allowing the portal to resolve this specification at runtime by automatically finding or generating relevant content.
The abstraction of the real content and the learning concepts is a key feature that supports the runtime selection of the most appropriate material. From the perspective of service composition this may be represented by the abstraction of the business goals and the actual services used to fulfill them, allowing for the dynamic selection and substitution of services. In both of these domains the issue of semantic interoperability comes into play. The abstract language describing the process and the metadata employed to describe the services need to be interoperable. This does not necessarily mean that they need to be identical, only that there needs to be some mechanism for mapping between the vocabularies used in each.

The services delivered to the user will be adapted based on the context information available to those services. This information may include pertinent details about the learner's learning experiences, including their content display preferences, learning style and prior knowledge. It is the role of the adaptive services, and the adaptive service composition environment, to take account of this information when adjusting their delivery of learning content and services. Context information about the physical environment and surroundings of the learner may also be considered important. For example, if the learner is engaged in a learning experience on a PDA it may be inappropriate to deliver a high quality video stream as the terminal device may not be capable of rendering that content to a sufficient standard. If the learner is situated in a library it may not be appropriate to delivery audio content to them. By utilizing abstraction techniques the learner's needs may be fulfilled with different types of learning material selected based on their requirements and the current situation.

This approach potentially facilitates the separation of responsibility for the personalized learning experience to many services. For example, one service may be responsible for producing a personalized course outline, expressed in terms of the learning concepts to be taught, based on the learner's prior knowledge. This personalized course model may be taken by a service responsible for selecting appropriate learning material candidates from the repositories available. A third service may be responsible for the delivery of an appropriate candidate for the current physical environment as well as monitoring learner feedback. The separation of responsibilities enables individual services to specialize rather than attempting (unsuccessfully) to be many things to many learners.

9. CONCLUSION

This paper has examined the area of context information and how it may be applied to adaptive service composition. Namely, it has looked at how context is defined, gathered and may be utilized by adaptive information services. A case study of context usage in adaptive eLearning was also discussed.

The principle of adaptive service composition was also covered, arguing the benefits of many discrete services combined, possibly at runtime, to fulfill the learner's requirements. Monolithic CMSs invariably lead to user (tutor or learner) dissatisfaction with some features. By combining appropriate context information about the learner with the facilities offered by many discrete services the learner's needs may be addressed in a truly personalized manner.

10. ACKNOWLEDGMENTS

This work is funded in part by the Higher Education Authority (HEA) of Ireland in the auspices of the M-Zones (Managed Zones) project [25]. The overall goal of M-Zones is to undertake fundamental research into novel management infrastructures to enable collaboration and management, between and within Smart Spaces. The key action in the project is to develop novel information and communications management technology to support dynamic, integrated management of participants, information appliances and smart space infrastructure.

11. REFERENCES

[1] [Abowd et al, 97a] Abowd, G., Atkeson, C., Hong, J., Long, S., Kooper, R., and Pinkerton, M. Cyberguide: A mobile context-aware tour guide. ACM Wireless Networks, 3:421–433, 1997.

[2] [Abowd et al, 97b] Gregory D. Abowd, Anind K. Dey, Robert Orr, and Jason A. Brotherton (1997). Context-awareness in wearable and ubiquitous computing. In ISWC, pages 179–180.

[3] [Abowd et al, 02] Abowd Gregory, Ebling Maria, Hunt Guerney, Lei Hui & Gellersen Hans-Werner, (2002), Context-Aware Computing, IEEE Pervasive Computing, IEEE

[4] [ADL] "ADL Sharable Content Object Reference Model", Version 1.2, http://www.adlnet.org/.

[5] [AICC] "AICC CMI Guidelines for Interoperability", Revision 3.0.1, Release 24 November 1999.

[6] [Blackboard] Blackboard Inc.: Blackboard Course Management System, Blackboard Inc. (2002) available online at http://www.blackboard.com/

[7] [Brusilovsky , 96] Brusilovsky, P.: Methods and techniques of adaptive hypermedia. In P. Brusilovsky and J. Vassileva (eds.), Spec. Iss. on Adaptive Hypertext and Hypermedia, User Modeling and User-Adapted Interaction 6 (2-3), 87-129.

[8] [Brusilovsky, 98] P. Brusilovsky. Adaptive educational systems on the world-wideweb: A review of available technologies. In Proceedings of Workshop "WWW-Based Tutoring" at 4th International Conference on Intelligent Tutoring Systems (ITS'98), San Antonio, TX, 1998.

[9] [Castro and Muntz, 00] Castro P. and Muntz, R. Managing context for smart spaces. IEEE Personal Communications,October 2000.

[10] [Casati et al, 00] Casati, F., Ilnicki, S., Jin, L., Krishnamoorthy, V., Shan, M. Adaptive and Dynamic Service Composition in eFlow. HPL-2000-39 20000406 External, March 2000.

[11] [Chakraborty and Joshi, 01] 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.

[12] [Chen and Kotz, 00] Chen, G. and Kotz, D. A survey of context-aware mobile computing research. Technical report, Department of Computer Science, Dartmouth College, 2000.

[13] [Conlan, 00] Conlan, O., and Wade, P.: Novel Components for Supporting Adaptivity in Education Systems - Model-based Integration Approach, ACM Multimedia 2000, November 2000.

[14] [Conlan et al, 01] Conlan, O., Hockemeyer, C., Lefrere, P., Wade, V., & Albert, D. (2001). Extending educational metadata schemas to describe adaptive learning resources. In Hugh Davies, Yellowlees Douglas, & David G. Durand (eds.), Hypertext 01: Proceedings of the twelfth ACM Conference on Hypertext and Hypermedia, pp. 161-162, New York: Association of Computing Machinery (ACM).

[15] [Conlan et al, 02] Conlan, O., Wade, V., Bruen, C., Gargan, M.: Multi-Model, Metadata Driven Approach to Adaptive Hypermedia Services for Personalized eLearning. In the Proceedings of Second International

Conference on Adaptive Hypermedia and Adaptive Web-Based Systems, AH 2002 (2002) 100-111.

[16] [Dey & Abowd, 00] Dey Anind K. and Abowd Greogory D. (2000), A Conceptual Framework and Toolkit for Supporting the Rapid Prototyping of Context-Aware Applications, Available: http://www.cc.gatech.edu/fce/ctk/pubs/HCIJ16.pdf [2002, Nov. 27]

[17] [Dey, 02] Dey Anind K. (2002), Understanding and Using Context, Available: http://www.cc.gatech.edu/fce/ctk/pubsPeTe5-1.pdf

[18] [EASEL] Educators Access to Services in the Electronic Landscape (EASEL). EC IST project 10051, http://www.fdgroup.com/easel/.

[19] [Ebling, Hunt & Lei, 01] Ebling Maria R., Hunt Guerney D. H. & Lei Hui (2001), Issues for Context Services in Pervasive Computing, Available: http://www.cs.arizona.edu/mmc/13%20Ebling.pdf [2002, Nov. 27]

[20] [Fink et al, 96] Fink, J., A. Kobsa, and A. Nill. User-Oriented Adaptivity and Adaptability in the AVANTI project in Conference "Designing for the Web: Empirical Studies", Microsoft, Redmond, WA.

[21] [IMS] IMS Global Consortium Inc., http://www.imsproject.com/.

[22] [Kass and Stadnyk, 92] Kass, R. and Stadnyk, I.: Using User Models to Improve Organizational Information. In Proceedings of the 3rd International Workshop on User Modeling. 1992, Dagstuhl, Germany.

[23] [Kobsa, 93] Kobsa, A., User modeling: Recent Work, Prospects and Hazards. In Adaptive User Interfaces: Principles and Practice, M. Schneider-Hufschmidt, T. Kühme, and U. Malinowski, (eds.). 1993, North-Holland: Amsterdam.

[24] [Mitchell, 02] Mitchell Keith (2002), A Survey of Context-Awareness, Available: http://www.comp.lancs.ac.uk/~km/papers/ContextAwarenessSurvey.pdf [2002, Nov. 27]

[25] [M-Zones] M-Zones (Managed Zones) http://www.m-zones.org

[26] [Piccinelli and Mokrushin, 01] Piccinelli G. and Mokrushin, L. Dynamic e-service composition in DySCo. 21st International Conference on Distributed Computing Systems Workshops (ICDCSW '01) April 16 - 19, 2001 Mesa, Arizona.

[27] [Rich, 89] Rich, E.: Stereotypes and User Modeling. In A. Kobsa and W. Wahlster (eds.): User Models in Dialog Systems, pp. 35-51. Springer, Berlin, Heidelberg.

[28] [Schilit et al, 94] Schilit, B., Adams, N., and Want, R. Context-aware computing applications. In IEEE Workshop on Mobile Computing Systems and Applications, Santa Cruz, CA, US, 1994.

[29] [Schilit, Adams & Want, 02] Schilit Bill N., Adams Norman & Want Roy (2002), Context-Aware Computing Applications, Available: http://www.fxpal.com/people/archive/schilit/wmc-94-schilit.pdf [2002, Nov. 27], IEEE Workshop on Mobile Computing Systems and Applications, IEEE

[30] [Schmidt et al, 99] Schmidt A., Beigl M., and Gellersen, H. There is more to context than location. Computers and Graphics, 23(6):893–901, 1999.

[31] [Su, Lee & Gerla, 02] W. Su, S. Lee, and M. Gerla (2000). Mobility prediction in wireless networks.

[32] [Want et al, 92] Want, R., Hopper, A., Falcao, V., and Gibbons, J. The active badge location system. ACM Transactions on Information Systems, 1992.

[33] [WebCT] WebCT: WebCT Course Management System, Lynnfield, MA, WebCT, Inc. (2002) available online at http://www.webct.com

[34] [Winograd, 01] Winograd Terry (2001), Architectures for Context, Available: http://hci.stanford.edu/~winograd/papers/context/context.pdf [2002, Nov. 27] HCI Journal, *Lawrence Erlbaum Associates, Inc, Mahwah, N*