Integrating History and Activity Theory in Context Aware System Design

Manasawee Kaenampornpan and Eamonn O'Neill

Department of Computer Science,

University of Bath,

Bath, BA2 7AY

United Kingdom

{cspmk, eamonn}@cs.bath.ac.uk

ABSTRACT

In this paper, we describe our context model as a design tool for developing context aware systems. Activity Theory is introduced as a potential approach for identifying and relating the elements that should be taken into account when designing context aware systems. We extend Activity Theory by adding the concept of history to produce the basis for our context modelling.

Keywords

Context, Activity Theory, history, pervasive computing

INTRODUCTION

Two of the factors that can impair the usability of mobile and pervasive systems are increased cognitive load on users attempting to multitask in busy environments, and the restricted input techniques typically available with both mobile and fixed devices. Usability can suffer particularly when there is a need for explicit input. Explicit input is input where the user tells the computer directly (e.g. by command-line, direct manipulation using a GUI, gesture or speech input) what he expects the computer to do, whereas implicit input is an action performed by the user that is not primarily aimed at interacting with a computer system but which such a system understands as input [10]. The need for explicit input may be reduced by increased use of implicit input. Therefore context awareness is an important concept for the usability of pervasive systems as it reduces the need for explicit input by taking advantage of changes in information relating to users, devices and environments. However, the research area of context history [2-4, 7, 11] is quite undeveloped and does not have well-established methods and techniques.

In order to derive principled design methods for developing context aware systems, we require system development processes, tools and techniques that take account of context. We must be able to develop systems that can determine implicitly the data that the user would otherwise enter explicitly. In our research, we have added the concept of history to Activity Theory [9] to provide a design tool to support the designers of context aware systems. Our extension of Activity Theory is used to provide guidance on what elements of context to take into account. It also supports the implementation process and both user- and system-driven adaptability at runtime.

ACTIVITY THEORY

From context classification systems reviewed in [7], researchers have classified context into different elements that have impact on users in performing their activities. Activity Theory is a philosophical framework used to analyse and model human activities. It was developed by the Russian psychologists of the former Soviet Union, Vygotsky, Rubinshtein, Leont'ev and others, beginning in the 1920s [9]. Vygotsky proposed that human actitivities are mediated through tools or instruments; this introduced the first generation of Activity Theory, modelled as a simple triangular structure of Subject-Tool-Object.

Engeström [5] proposed a more comprehensive model of human activity (see Fig. 1). This model was based on the work of the first generation of Activity Theory and on the idea of the general structure of animal activity, consisting of the individual, natural environment and population. Engeström supported the main concept of Activity Theory that individuals' actions are influenced by their socio-cultural context and therefore cannot be understood independently of it [5]. The full triangular structure of human activity that was introduced by Engeström suggests that the relationship between the subject and the community is regulated/mediated by rules and that the relationship between the community and the object is regulated/mediated by a division of labour. Activity Theory maps the relationships amongst the key elements that it identifies as having an influence on human activity. With Activity Theory, we have a simple standard form for modelling human activity (see Fig. 1).

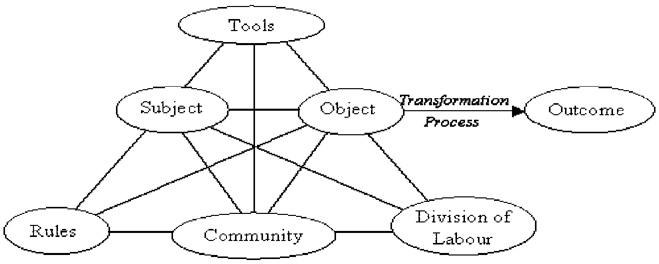

Fig. 1. Full structure of human activity introduced by Engeström et al. [5].

In modelling context for context aware system design purposes, we argue for using a simple standard form to model the aspects of human activity that are associated with key elements of context. Although a simple standard form cannot represent the full richness and complexity of human activity, it does not have to. As humans, we cannot and do not form complete models of other humans' context, especially with regard to their internal goals and intentions. Despite using partial and simplified models, we manage to communicate and collaborate with our fellow humans very effectively and efficiently. From time to time we do get it wrong, for example, misinterpreting another person's intention or meaning. We then invoke repair mechanisms and feed the information generated through this experience into our future models. Since humans manage so well with relatively simple and partial models of other humans' goals and activities, it is both unreasonable and unnecessary to demand more of computer-based context models.

HISTORY

Although Activity Theory captures the key elements of human behaviour, it only captures information about the user's current situation or context and the outcome when the current activity is performed. It does not provide an adequate account of a user's current object or intention, or of the user's past actions and contexts. People often refer to experiences in the past while performing their current activity, using such experiences to guide their current actions. Chalmers [2] notes a range of research that refers to activity as an ongoing temporal process of interpretation. He found significant potential in making more use of the past in context aware system design.

History is a crucially important part of context. A few previous context awareness projects have considered time as context. However, they have typically looked at time simply as current time that can be sensed from the device. For example, they compare current time to the user's timetable and provide support for the user's current task in her timetable [1, 6].

CONTEXT MODEL

Webster's dictionary [8] defines time as "a nonspatial continuum that is measured in terms of events which succeed one another from past through present to future". It defines history as "a treatise presenting systematically related natural phenomena".

Time gives us a means of referencing the occurrence of events; therefore by adding a timeline to the Activity Theory model, we can represent the history element in our context model (see Fig. 2). The timeline includes not just current time, but also past time (which contributes historical elements to the context) and future time (which allows for prediction of users' activities from the current context).

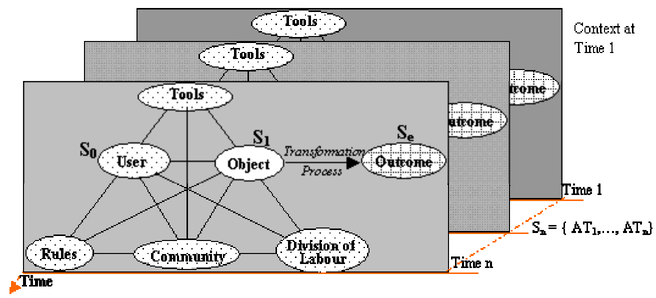

Fig. 2. Extending Activity Theory to represent history.

History is modelled as an abstraction over a set of states in the past. Each past state is represented as an Activity Theory model, which captures the context of activities at that time. This information includes the initial state (**S_0**), intention or Object in Activity Theory (**S_1**), and outcome or end-state (**S_e**) of the activity. The initial state (**S_0**) includes the current information about user, tools, rules, community and roles. The intention (**S_1**) models information about the user's current goal, i.e. what the user is trying to achieve. This information about user intention (**S_1**) can be inferred from the history of context (**S_n**) and the initial state (**S_0**). Once the user has performed the activity, we have information about the outcome (**S_e**). Then, the initial state (**S_0**), intention (**S_1**) and outcome (**S_e**) become part of the history of context and will be used to help infer the user's intention or goal in future situations.

APPLYING THE CONTEXT MODEL

During Design

We propose the context model illustrated in Fig. 2 as a design tool to aid the designer in building an understanding of context. It helps make design issues explicit and forms a basis for design choices. It also encourages the designer to focus on aspects of the system affecting usability.

The context model is used to generate a checklist for the designer to focus on what should be taken into account as context, derived from the elements in the context model. The first step for the designer is to expand the key elements in the context model into sub-elements of information

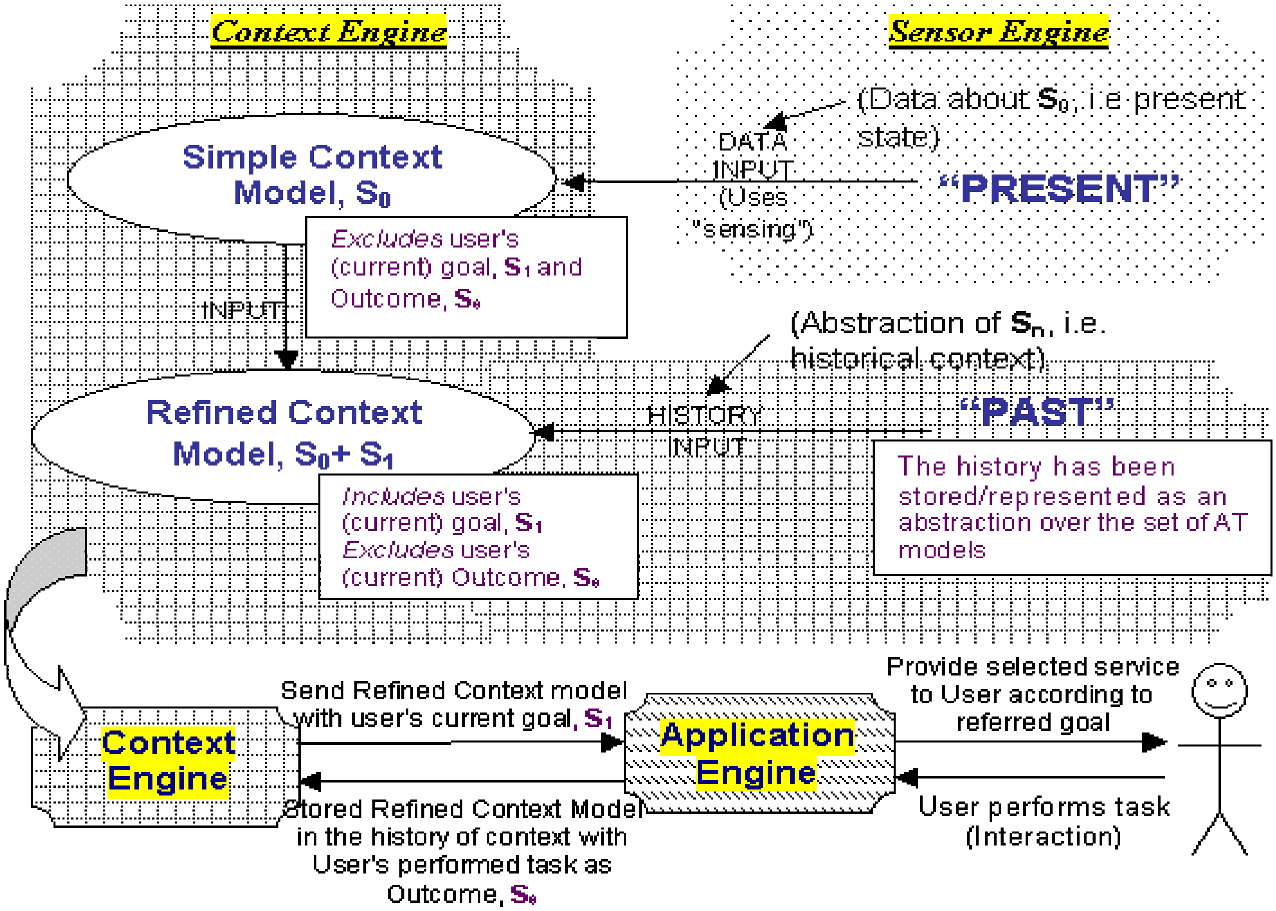

Fig. 3. Information processing in context aware system.

about context, for example information about user's ID, device, location are sub-elements of the User element. The second step is to generate rules to reason about the context according to the relationships between each element in the context model. (These rules are subsequently used to develop a rule-based system, at the implementation stage of the system development process.) The third step for the designer is to check the availability of technologies that can capture the context information in the sub-elements in the checklist.

During Implementation

Fig. 3 illustrates how the system infers the user's current objective, goal or intention. The system first collects information on each element in the context model. This information may be collected from sensors or databases. At this stage, the context model is used as a guide for the system to what types of information to take into account. Secondly, this information is used to model the context of the user's current activity ($\mathbf{S_0}$) based on the rules that the designers have set (and which may subsequently have been adapted at runtime). This simple context model does not include the user's intention or goal ($\mathbf{S_1}$), as we cannot sense such information. Thirdly, the context model references the history ($\mathbf{S_n}$), which records the user's context in performing her activities in the past, in order better to infer the user's current objective. This results in a refined context model that includes a model of the user's current goal. Then the user's current goal is used to identify the appropriate service to support the user's current goal. Moreover, the user's interaction is monitored and recorded as the outcome of the current context model.

Fig. 3 shows that the context model also underlies an implementation architecture that has clear separation between a Sensor Engine (that controls input from different sensors and transforms them into appropriate data input for the context engine), a Context Engine (that matches data input into elements in a context model and infers the user's current goal in the context model by referring to the history of context) and an Application Engine (that selects a service to support the user's current goal by referring to the history of support). Hence, changing sensors or adding/removing application services can easily be done after the system is implemented.

During Run-Time

The system supports adaptation during runtime by both the system and the user (see Fig. 3). For example, the system can downgrade or remove from the reasoning process in the system the sub-elements of context that have not been used for a certain period of time.

It also supports the user during runtime by giving her a structural understanding about the system; what the system is taking into account as context and how the system reasons about the context. With this understanding of the system, the user is better able to adapt the system according to her requirements. For example, the user can change the rules that have been set in the system when the system does not perform optimally.

BENEFIT OF USING HISTORY

During run-time, the input from sensors may be inaccurate or missing. Fig. 3 shows that input from sensors is processed to fit into the simple context model. The simple context model of the current situation is then used to infer the user's current goal, by referring to the history of context. The process of referring to the history can reduce sensor input problems because the current context is referred to the history as a whole set of context in a model, not as a single input value from the sensor. Therefore, if an input from a sensor is inaccurate or missing but the rest of the values in the context model match the history then we will still get the best-matched current goal.

Similarly, in the application engine, the user's current goal is used to match with the history of services that the application engine has selected to support the user in the past. The history of selected services also holds information about user interaction after the selected service was provided to the user. Therefore the system can use this information to improve performance in selecting the services to support the user's current goal.

CONCLUSIONS

In this research, we considered Activity Theory as an appropriate framework to provide a comprehensive context model that includes the key elements of context that have an influence on a user's diverse activities in a mobile and pervasive computing world. Activity Theory also identifies the relationships between each element in the model so that these relationships may be applied during the development of a context aware system. We identified in this paper that history is important for humans while they are performing their current activity; therefore, we have extended Activity Theory to capture the concept of history in our context model. This model can then be used a design tool for developing context aware systems that reduce the need for explicit input from users. With a design based on a sound model of context and the capacity for runtime adaptability based on past performance and current preferences, such a system will go some way towards achieving the goal of reducing the need for explicit user input and thereby increasing the usability of mobile and pervasive systems in situations of high cognitive load and constrained input devices.

REFERENCES

1. Agarawala, A., Greenberg, S. and Ho, G. The Context-Aware Pill Bottle and Medication Monitor *The Sixth International Conference on Ubiquitous Computing*, Nottingham, England, 2004.
2. Chalmers, M. A Historical View of Context. *The Journal of Collaborative Computing: Computer Supported Cooperative Work*, *13* (3-4). 223-247.
3. Chen, G. and Kotz, D. A Survey of Context-Aware Mobile Computing Research, Dartmouth College, Department of Computer Science, UK, 2000.
4. Dey, A.K. and Abowd, G.D. Toward a Better Understanding of Context and Context-Awareness, Georgia Institute of Technology, Atlanta, GA, USA, 1999.
5. Engeström, Y., Miettinen, R. and Punamäki, R.L. (eds.). *Perspectives on Activity Theory*. Cambridge University Press, 1999.
6. Hertzog, P. and Torrens, M., Context-aware mobile assistants for optimal interaction: a prototype for supporting the business traveler. in *The 9th international conference on Intelligent user interface*, (Funchal, Madeira, Portugal, 2004), ACM Press, 256-258.
7. Kaenampornpan, M. and O'Neill, E., Modelling context: an Activity Theory approach. in *Ambient Intelligence: Second European Symposium, EUSAI 2004*, (Eindhoven, The Netherlands, 2004), Springer, 367-374.
8. Merriam-Webster, Merriam-Webster Dictionary. Merriam-Webster, Incorporated, Last Access 17 Feb 2005, (2005) www.m-w.com
9. Raeithel, A. Activity Theory as a Foundation for Design. in Floyd, C., Zllighoven, H., Budde, R. and Keil-Slawick, R. eds. *Software Development and Reality Construction*, Spinger Verlag, 1992.
10. Schmidt, A. Implicit Human Computer Interaction Through Context. *Personal Technologies*, *4* (2). 191-199.
11. Schmidt, A., Beigl, M. and Gellersen, H.W. There is More to Context Than Location. *Computers and Graphics*, *23* (6). 893-901.

AUTHOR BIOGRAPHIES

Manasawee Kaenampornpan (Jay) received her B. Eng. in Computer Systems Engineering from University of Warwick in 2001 and her M. Sc. in Advanced Computing: Global Computing and Multimedia from University of Bristol in 2002. Currently she is a PhD student in Human Computer Interaction Group at University of Bath under her supervisor, Dr. Eamonn O'Neill.

Eamonn O'Neill is a senior lecturer in the Department of Computer Science at University of Bath. He is a researcher in the field of Human Computer Interaction. He is the leading researcher in "Cityware: Urban design and pervasive system project", commencing in October 2005.