An Activity-Based Approach to the Design of User Assistance in Intelligent Environments

Christoph Stahl[1], Dominik Heckmann[2], Michael Schneider[2], Alexander Kröner[2]

[1]Dept. of Computer Science, Saarland University
stahl@cs.uni-sb.de
[2]DFKI GmbH, Saarbrücken
{heckmann,schneider,kröner}@dfki.de

Abstract. Designing user assistance systems in intelligent environments poses many challenges. The system has to provide useful support for its user in everyday situations, while keeping the user interface as simple as possible. This requires a well-founded understanding of the user's needs, and deep knowledge of pervasive human-computer technology, such as RFID sensors and computer vision. In this paper, we present a new design method, based on Activity Theory, and a toolkit that supports the designer of ambient intelligence systems. The toolkit integrates a geometric location model with a symbolic activity model, so that the typical activities of the user can be analyzed with respect to the environment in order to identify useful assistance features. Furthermore, the system supports the designer to make the necessary decisions for the instrumentation of the environment, i.e. which sensors to use and where to place them.

Introduction

Mobile and ubiquitous computing raise new opportunities for assistance systems. If computers are everywhere, they can assist not only in office tasks but also in everyday life situations. Much effort is aimed towards elderly people in our aging society who need care and immediate help in case of an emergency. The goal of ambient assisted living projects is to instrument their homes with sensors, recognize their activity, reminding them on everyday tasks if necessary (especially in case of dementia disease) and to alert a doctor in emergency situations.

However, as we envision new assistance systems with lots of interesting functions, we should be aware of the problems that their user interfaces might pose to their users. The perceived saving of time and attention has to be bigger than the necessary effort to learn and operate a new interface. Ideally, a smart system with access to sensors in the environment and fitted with artificial intelligence would automatically recognize the needs of the user. For example, vision based sensors can be used to recognize the user's mood from the facial expression, and everyday objects can be augmented with accelerometer sensors to recognize certain gestures. If combined, readings from multiple sensors allow the system to infer the actions and high-level activities and goals of the users. Based on this knowledge of the user's behavior, so-called intelligent or instrumented environments are able to foresee when user assistance is needed, and the system can proactively offer assistance in the sense of an agent. This is convenient for the user, who does not necessarily have to know how to explicitly interact with the system or how to use certain system functions. In the best case, the user will always receive the information or service that is required without having to ask for it.

However the design of such intelligent assistance systems requires a very precise model of the user's needs in order to implement useful features and to avoid undesired or disturbing system behavior. Furthermore, sensors and output modalities have to be chosen with great care. Finally, their deployment in the environment has to be planned where especially geometric details such as viewing angles of cameras and displays have to be considered.

In this paper, we present a new design method and a toolkit that supports the designer of ambient intelligence in their tasks. The toolkit is based on Activity Theory, which will be introduced in the following section, and integrates the activity model with a geometric location model.

Activity Theory

Activity theory has been introduced through the Russian psychologist Aleksey Leontiev (1904-1979). Leontiev's basic notion of activity is based on a triple made of *subject*, *activity*, and *object*. In contrast to object-oriented analysis (OOA) thinking, where objects are the top-level entities and relations are identified between them in successive steps, activity is here proposed as the basic unit of analysis providing a way to understand both subjects and objects. Since the theory is historically rooted in the investigation of the development of the human mind, the *subject* always refers to a living thing, which has either basic (animal or human) *biological needs*, such as the need for food to survive, or more abstract human *psychological needs*. The object might either be something that directly satisfies the need, such as food, or some tool that indirectly helps to acquire the desired object, such as an axe or computing device.

Thereof results the principle of *hierarchical structure of activity* as depicted in Fig. 1. An object that directly meets a certain need has the status of a *motive* that motivates the activity of the subject, whereas all secondary objects are considered as *goals* towards this motivating object. The steps which are necessary to achieve these goals are called *actions*, which in turn can be refined on a lower level through *operations*. They identify internalized routine processes which establish certain *conditions* that are required to execute an action, and people are usually not aware of them. A single activity can consist of multiple actions, and each action can be made of multiple operations. Kaptelinin and Nardi [1] give a good introduction to these terms and discuss how Activity Theory can contribute to the field of HCI.

Activity	**Motive**
Actions	**Goals or Aims**
Operations	**Conditions**

Figure 1. The hierarchic structure of human activity according to Leontiev's Activity Theory.

Design Method

The concepts of activities, actions and operations make a good instrument to describe and analyze a given situation in order to design a user assistance system. Our proposed toolkit supports the designer through the process in five steps:

1. Geometric Location Modeling
2. Activity Modeling
3. Identify opportunities for User Assistance
4. Choose and Place Sensor Input and Actuator Output
5. Simulate and Evaluate Virtual Instrumented Environment

The process starts with the geometric modeling of the environment in 3D based on a ground floor plan as first step. The Yamamoto (**Y**et **A**nother **Ma**p **Mo**delling **To**olkit) software [2] has been originally developed to model multi-level buildings for indoor navigation applications that assist pedestrians in their wayfinding tasks. In order to support the design of instrumented environments, we have added features to model furniture objects, shelves, artifacts (e.g. home appliances), and interaction devices (i.e. optical devices like displays, projectors, and cameras).

In the second step, activities are modeled using our newly developed Activity Editor application, which provides a tree-grid table that well supports the hierarchic structure of activities, actions, and operations. Each entry basically represents a triple of subject, artifact, and objective (need). This classic notion of an activity is extended by virtual artifacts (information sources, represented by URLs), location, and time. Instead of entering plain text, the UbisWorld [3] ontology provides the necessary concepts which can be entered into the table through a drag and drop operation. The ontology defines a vocabulary of names for almost everything, which encode multilingual labels into unique identifiers that conform to the strong syntactic limitations of URLs and Semantic Web languages (e.g. OWL, RDF). This mechanism allows the toolkit to directly link the symbolic representation of artifacts, locations, and devices to their graphical counterparts in the virtual environment model. Hence the described actions and situations can be easily browsed and visualized in 3D.

Once the scenario has been modeled in this way, opportunities for useful kinds of user assistance become clear: Which information might be needed? Which actions are likely to be forgotten and need a reminder? Which actions and operations are difficult to perform and might require physical assistance?

Based on this analysis the designer can define assistance actions that are performed by the system and has to choose interaction modalities and devices, including sensors. The ontology provides a collection of input sensors and output actuators, which can be instantiated and included in the virtual environment model in order to plan and validate their actual physical placement and viewing angle through simulation.

We have applied our toolkit to the instrumentation of a kitchen environment that assists the user in recording and reproducing cooking recipes.

Use Case: Assistance in the Smart Kitchen

The Smart Kitchen captures the user's context and action during the cooking in form of video and sensor data from the instrumented environment. The system is equipped with various sensors, cameras, and networked appliances capture the user's actions and context during cooking. Multiple wide-angle cameras record an audio/video stream of the cooking session. Each movable object in the kitchen is tagged with a passive RFID transponder. RFID antennas are mounted discretely at important locations in the kitchen and allow detecting if an ingredient or tool is placed on such a location. Finally, Networked kitchen appliances (stove, oven, and fridge) report about their state and usage. The Smart Kitchen is equipped with a wide-screen TFT panel to display the expert's cooking process. The screenshot (Fig. 2) shows the virtual environment model, part of the ontology, and the activity model.

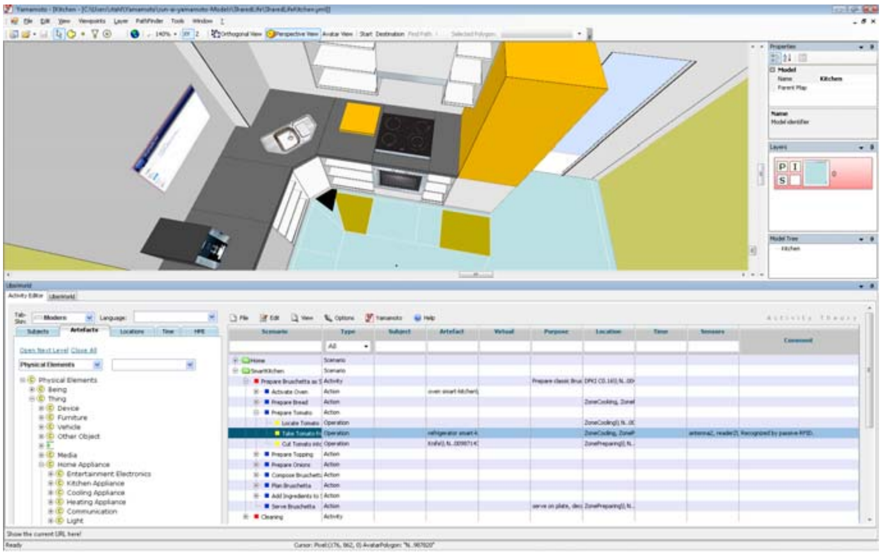

Figure 2. The design toolkit integrates a 3D model (upper frame) with a tabular activity model (lower frame), that uses ontological concepts (lower left).

References

[1] Kaptelinin, V. and Nardi, B. (2006). Acting with Technology – Activity Theory and Interaction Design. The MIT Press, Cambridge, Massachusetts.

[2] Stahl, C., & Haupert, J. (2006). Taking Location Modelling to new Levels: A Map Modelling Toolkit for Intelligent Environments. In: M. Hazas, J. Krumm, and T. Strang (Eds.): LoCA 2006, 74 – 85. Springer. 3987.

[3] Heckmann, D. (2005). Ubiquitous User Modeling. DISKI 297, Akademische Verlagsgesellschaft Aka, Berlin.