Framework for Visual Analysis of User Behaviour in Ambient Intelligence Environment

[1]Ivo Maly, [2]Jan Curin, [1]Pavel Slavik and [2]Jan Kleindienst
[1]Czech Technical University in Prague, Faculty of Electrical Engineering, Czech Republic;
[2]IBM Czech Republic, Voice Technologies and Systems

1. Introduction

Ambient intelligence is a rapidly developing field that is led by the necessity of ubiquitous computing. People are surrounded by computers in most of their daily activities; i.e. in cars, homes, offices and public areas, and supplied by various services on mobile phones, PDAs, or interactive kiosks. There are several projects that demonstrate potentials of ubiquitous computing, e.g. in a museum (Gallud et al., 2007), in a home (i2home, 2009), or in an office (Waibel, 2009).
In our case, the ambient intelligence environment consists of two components, a mechanism for detection of user behaviour and an application that supports specific user scenario or set of scenarios (ambient intelligence applications). Along with increased complexity of the systems and user workflows in an ambient intelligence environment, the usability and functionality of both components must be ensured. Traditional approaches of usability testing are suitable for testing individual components in an isolated manner. However, usability of the whole system can significantly differ from the usability of individual components. A reason for testing of individual components is the inability to setup a complete environment of supporting technologies as they are still under development or yet non-existent in early phases of system design. Unfortunately, these phases are the ones where a correct design decisions are crucial, because they influence further development of the whole system.
In this paper we introduce a framework for testing applications in ambient intelligence environments. The concept is derived from the methodologies of desktop application usability testing combined with evaluation of user behaviour in an ambient intelligence environment. It is based on our experience with user behaviour recording, which is visualized and investigated mainly in a virtual environment. The visualization serves as a tool for analyzing complex situations and detection of usability issues in ambient intelligence applications.

2. Motivation

Usability testing methodologies for desktop applications are quite well known and well described e.g. by (Rubin, 1994) or (Nielsen, 1994). The methodologies typically use observation of the test participant, while solving a task from the task list where the tasks use the tested application. Indirect observation is recorded and can be replayed several times. Typically a small amount of participants (4-12) are tested and mainly qualitative issues are evaluated, interpreted, and used for suggested improvements. There may also be other data sources involved in application usability testing; typically application log files (Ivory & Hearst, 2001), where information about user interaction with application is stored. Information from such sources can help to thoroughly analyze particular parts of the application or the test.

Many modifications of this methodology have appeared. One modification that is significant for our ambient intelligence environment testing methodology is remote usability testing (Paterno et al., 2007). This methodology usually does not use observation of participant interaction but rather derive usability issues from user interaction data generated by application. This approach is suitable when observation is not possible due to technical limitations, e.g. test participant and usability expert are at different places. An advantage of the desktop application in this methodology is that we can quite accurately estimate context of use, based on target user group knowledge. This allows us also to focus more on the observation of user interactions, supplemented with user thoughts acquired by the thinking aloud technique (Rubin, 1994).

On the other hand, applications for ambient intelligence environments are much more context sensitive. Context can change several times during a single test and therefore we have to give the evaluator all the important context information. Observation of the user in such an environment is also problematic. Analysis from the recording from observation of participant interaction can be complicated when more cameras are used in a complex test setup. Moreover analysis of video recording is very time consuming. These issues can be solved by transformation of the participant behaviour and interaction into the visual representation (e.g. virtual environment) that is used for further test analysis and application usability study. There are also several other motivations that support the needs to be solved in order to improve process of user behaviour analysis.

The first motivation is the ability to analyze huge amounts of data that is gathered from the tested application and from the supporting infrastructure of the ambient environment. It is very complicated to analyze the data in its raw format such as a text file. The visualization in the virtual environment can be one of the suitable data transformations.

The second motivation is unification of different test data for direct comparison. Traditionally, we can face to the problem of comparison of data from ambient environment sensors (e.g. time snapshots of the ambient intelligence environment state), video recording from environments where no other sensors are available and user scenarios created artificially (e.g. text description). Therefore we should be able to transform all those data types to one virtual environment.

The third motivation is connected with ethical issues. The process of usability testing obeys strong ethical codices, which also keep to the laws that are part of the legal system.

Typically, it is not allowed to publish a video of the user from the test, where he/she can be recognized, without his/her permission. In desktop application usability testing the problem can be easily solved. The video from a desktop application focuses on the user interaction with the application, which means that the application is the object being recorded and the user usually does not appear in the video (the user can be recorded using another camera). In the case of mobile application usability testing, the user is part of the environment and his/her interaction and behaviour is being investigated. Such a video recording is then not usable as material for application usability issue presentation. Instead of making the user anonymous in the video recording the user can be made anonymous by the process of data transformation into the virtual environment where the user is represented in the form of avatar.

3. Related Projects

There are several projects that also deal with evaluation of ambient or ubiquitous applications. Project UbiWise (Barton & Vijayaraghavan, 2002) tries to simulate ubiquitous environment equipped with devices that communicate with each other. It focuses on simulation of hardware and low level software of the ubiquitous environment. Another project that has similar motivation as UbiWise is project UbiREAL (Nishikawa et al., 2006). In this article the authors present a tool for testing the correctness of virtual or real UPnP (Universal Plug and Play) protocol based appliances connected into a network. The authors focus on the presence of the appropriate appliance, controller functionality and logic instead of the user behaviour. Project 3DSim (Nazari Shirehjini & Klar, 2005) is tool for rapid prototyping of ambient intelligence applications. This tool uses also UPnP protocol (Nazari Shirehjini, 2005) for inter device communication. It is similar to both UbiWise and UbiREAL projects and it has some advantages like photo-realistic impression of visualization and support simple visualisation of human interaction in the environment based on recognized data. Project P-VoT (Seo et al., 2005) is component based interactive toolkit for virtual environments to quickly build a virtual ubiquitous environment. This tool allows rapid creation of the environment with possibilities of detailed device logic specifications.

We have identified two reliable pervasive computing systems which may serve as context acquisition and modelling platforms for multimodal services: already mentioned UbiREAL and functionally similar Context Toolkit (Dey et al., 2001). But in contrast to these systems, which integrate contextual information directly from various sensors, we needed a system which relies on information provided by more complex perceptual components, i.e. more complex context-acquisition components such as person trackers and speech recognizers. A body tracker might be at once capable of detecting location, heading and posture of persons, identifying them, and tracking subjects of their interest. Such scenarios led us to separate the perceptual components layer from the layer that deals with higher abstraction – the situation modelling. Defining and modelling situations based on a wide range of context-acquisition components was not supported by other environments such as UbiREAL and Context Toolkit, so we decided to implement a new framework.

4. Overview of the Framework

4.1 Introduction

Evaluation of applications in ambient environment is a complex issue. Therefore in next sections we will introduce framework that was developed to encapsulate all tasks necessary for analysis of user behaviour as part of usability evaluation of applications in ambient intelligence environment. Scheme of the framework is in the Fig. 1. Data from detection mechanism, that consists of sets of sensors (e.g. noise level detectors) placed in the ambient intelligence environment, are imported to the framework's internal data storage either directly or through the Testing of detection mechanism accuracy task. The data can be also created and edited in User scenario creation and editing task. These two tasks are depicted by rectangle with rounded corners and represent group of tasks that edit raw data. Once the data are in internal format they are persisted or immediately used for visualization in next two tasks. These two tasks are Evaluation of single user scenarios and Evaluation of cooperative scenarios. They represent high level task that analyze the data as user behaviour in the ambient intelligence environment. In next chapters we will describe each task in detail.

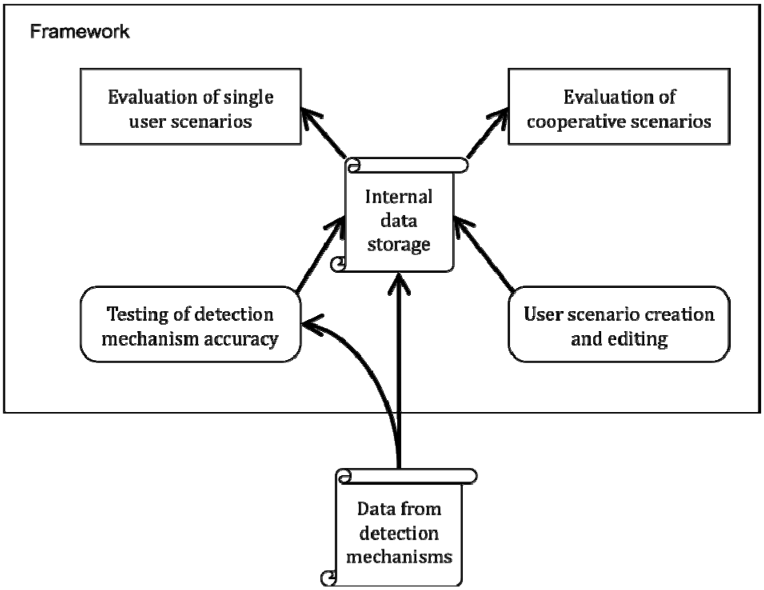


Fig. 1. Scheme of the framework.

4.2 Testing of Detection Mechanism Accuracy

Testing of detection mechanism accuracy is the lowest type of test. Aim of this test is analysis of accuracy of particular sensors and algorithms that process data from those sensors. Ambient intelligence applications strongly rely on those detection mechanisms and their algorithms and incorrect output of the detection process may highly influence behaviour of the system and usability of the application for the user.

Typical use case is evaluation of data from one or several detection mechanisms and comparison of them with either video recording or with manually revised data usually extracted from video recording. Output of this task is either report about detection mechanism accuracy or correction of data produced by the detection that are then used for evaluation of user behaviour or application usability in next types of evaluation.

4.3 User Scenario Creation and Editing

At the early stage of testing the sensors are not always available. Therefore it is not possible to collect data from them and analyse user behaviour in the environment. Therefore the framework must contain task where the data are created and edited.

Typical use case is creation of single user scenario from external formats (e.g. audio/video recording), creation of user data by expert creation of cooperative scenario from external formats (e.g. audio/video recording) and enhancement of the data from detection mechanism with additional information. Such information may be for example interaction with objects that is not detected by any detection mechanism. Execution of this task is demonstrated in section 7.1. Output of this task is set of scenarios.

4.4 Evaluation of Single User Scenarios

This type of evaluation is the simplest case of usability evaluation of the application. We investigate behaviour of the user according to the predefined scenario. We should compare the user behaviour with pre-recorded data created by expert, in order to find deviations from the expected behaviour. Expert data may be gathered directly from the ambient intelligence environment and eventually polished in task testing of detection mechanism accuracy. Alternatively they can be created by the framework in task user scenario creation and editing, where the simulation of the user interaction in the ambient intelligence environment is performed by expert.

There are several possible use cases how to perform this task. The first use case is simple evaluation of one user. The second type of use case is comparison of one user with the data created by expert. The third use case is comparison of several users between each other. Finally fourth use case is comparison of several users and data from expert between each other. Execution of this task is demonstrated in section 7.3.

4.5 Evaluation of Cooperative Scenarios

Cooperative scenarios are the most complicated scenarios for evaluation in all environments. Goal of this test is to evaluate whether the application correctly supports cooperation of individual users in their scenario. Currently the only use case supported is evaluation of one cooperative session. Execution of this task is demonstrated in section 7.2.

5. Visualization of User Behaviour

5.1 Introduction

When performing various user tests and other experiments that deal with user behaviour we always obtain large data sets. These data should be evaluated and some conclusions should be drawn in order to improve the design of particular user interface. Due to the large amount of data it is mostly impossible to perform analysis of these data without deeper insight (to understand mutual relations between various variables or between data subsets in the given data set). Solution to this problem is to use an appropriate visualization technique by means of which we can get a deep insight into data investigated (and thus to make some relevant conclusions).

There exists a large amount of various visualization techniques by means of which it is possible to visualize data (or information of various kind) – these techniques are of interest in a special branch of computer graphics, particularly, information visualization (Spence, 2000). In our research we had to select and modify some approaches in order to be appropriate for our purpose that is visualization of user behaviour in ambient intelligence environment.

5.2 Survey of Visualization Techniques

At first we have to define what kind of activity we are going to investigate and visualize. As our research has been concerned with user interaction in ambient intelligence environment we had to investigate (and visualize) three main aspects of user behaviour:

- visualization of user interaction with devices (log file with user activities when solving a particular task).
- visualization of the user movement in the scene (visualization of the user movement in the scene gives us a global picture about the activities of the user)
- visualization of the user movement and behaviour (such a visualization offers a complex picture about the user when solving particular task in ambient environment)

In Fig. 2 we can see that these three aspects of user behaviour form hierarchy – here we can speak about a sort of level of detail in which we would like to investigate the data obtained. In (Mikovec et al., 2007) we showed that this concept corresponds with three types of visualization. In our framework we also choose appropriate visualization method when investigating particular user behaviour. In such a way we can analyze both individual approach to the solution of task (that consists of several subtasks) and movement of the user in the scene where the task should be solved. In further text these three visualization methods, that are suitable for visualization of three data types given above, will be presented.

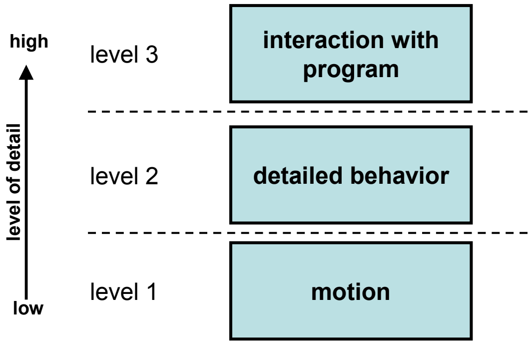


Fig. 2. Level of detail of user behaviour visualization.

In all visualization methods (levels) we tried to analyze and describe the user behaviour based on data set, which was acquired through observation of user interaction with application (stored as interaction log file). Data set is created by means of collected data that reflect the user activity when the user performs some typical tasks. These activities (performing certain task or collection of tasks) are performed in the framework of usability testing (Rubin, 1994) (the user interface designed is tested by several of users who perform some typical tasks). The data about the performance of users are collected using various methods – mostly observation of the user or recording the user activities by means of various tools (recording the individual interaction steps during particular task solving etc.). In order to understand such a complex data we need to have a proper visualization tool that is interactive and allows us to select proper view on the data.

For the Level 3 we can use visualization in form of timelines described in (Harrison et al. 1994) and (Malý & Slavík, 2007). These visualizations allow visualizing data both from standard usability testing and also from ambient intelligence environment. The timeline (Fig. 3) is presented as a sequence of annotations ordered according to the start time of the annotation. Each annotation represents one activity of the user during the usability testing. Annotations are grouped into categories based on the severity of the annotated problem. The severity is represented by different colour of the annotation – red is more complicated for the user, white is easier for the user). The visualization also shows annotations together with task tree (Paterno et al., 1997) (that describes in a formal way the structure of the task performed) in one view. The task tree in visualization shows by green rectangles at which time certain activity (e.g. subtask) begun and when this activity was finished.

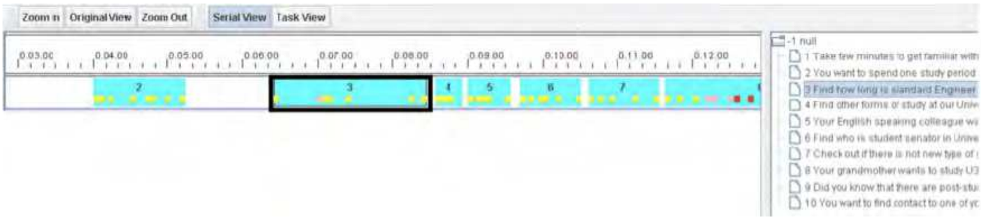


Fig. 3. Timeline visualization of user interaction. Colour of annotation represents severity of the problem.

As we are in our research concentrated on a specific environment – ambient environment – we should have some idea about the user motion in this environment. For the sake of simplicity we can imagine that a user walks in an exhibition hall, in a gallery, in a museum etc. When following her/his path we can discover some patterns (like concentration on specific areas – where e.g., are exhibited topics of her/his interest like some products, pictures etc.). In such a way it will be possible to discover attitudes of individual users (this may help to define groups of users that share common interests etc.). This kind of evaluation of the user behaviour, representing Level 1 in Fig. 2, is based on visual analysis of the acquired data. The user is trying to get from the position A into the position B. What is interesting is the investigation of strategy for selection of the trajectory that has been chosen by the user. This information can give important information about the user background on which the selection of strategy was based on. The visual analysis, see Fig. 4, results in determination of spots in ambient environment where the user spent more time or where the user appears frequently. By means of proper visualization tool it is possible to identify these spots in a very short time.

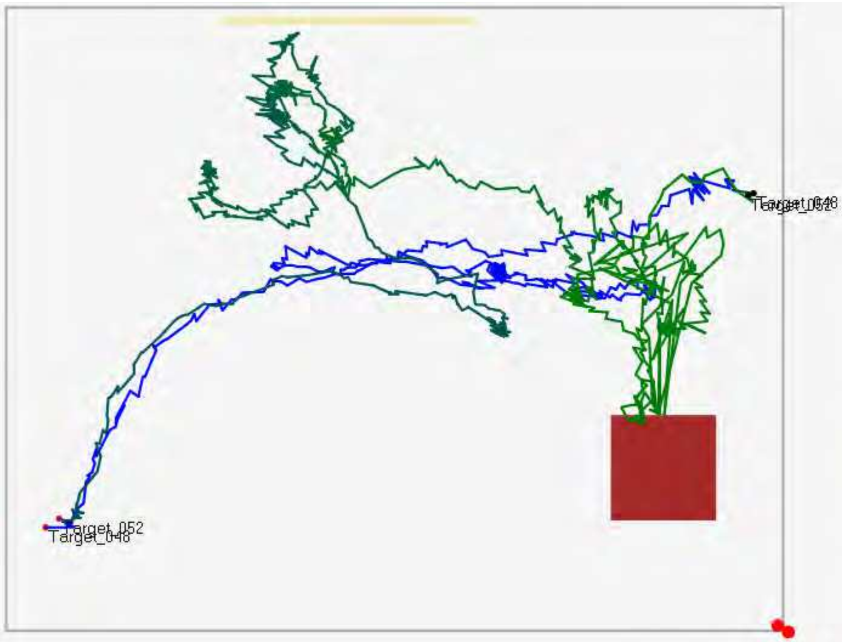


Fig. 4. Visualization of the movement of the user in the room.

Fig. 5. 3D visualization of the user interaction in the room.

The fact that the third dimension gives us many advantages when visualizing data about the user behaviour resulted in development of a new visualization technique, which represents Level 2 in Fig. 2. This third visualization technique allows us to visualize user movement in an ambient environment – e.g. in a smart room. The data for visualization are from real sensors or they are manually transformed from video recordings into simulated user scenario that is interpreted in a simulator. The output of the simulator is visualization of the user behaviour in 3D. Such a visualization is performed in virtual environment where we can manipulate both with the virtual environment (zooming, rotations with camera, …) and with data visualized (data about motion or behaviour of the user). Thanks to that we can get much better idea about user's intentions, habits, behavioural patterns, etc. Such an approach provides higher (semantic) abstraction in the situation modelling module, and presents it to the user graphically in a 3D scene, see the Fig. 5.

When summing up the properties of the visualizations described it is possible to say that the visualizations presented create a scalable system that allows user to visualize complex data obtained in the framework of usability testing. The possibility to select proper visualization makes the evaluation of data collected more flexible and more efficient (it is possible to evaluate data on the proper level of detail).

6. Applications Used in the Framework

6.1 Introduction

Our framework is based on our experience with development and usage of two applications for analysis of user behaviour in ambient intelligence environment. The first application is

SitCom - Situation composer. The second tool is USEd - User Scenario Editor. Description of both tools is in next sections, followed by experiences with these tools in several use cases.

6.2. SitCom – Situation Composer

SitCom (Fleury et al., 2007) serves as a simulation runtime. i.e. the core tool supporting the whole development cycle of multimodal perceptual systems.

6.2.1 Smart Environment Architecture

Before we introduce the SitCom tool, we should describe basic terms and abstraction architecture of smart environments. Sensors deliver their information in a streaming fashion, as do video camera for video data or noise level detectors for a given value of the noise level. The perceptual components, which do some combination of the sensor data to render it at a higher semantic level, still propagate this information, after processing, in a streaming fashion. A body tracker will update the number of detected bodies, and their locations. A room noise classifier will deliver the new label representing a person in the room as soon as it has processed the information from sensors. In that sense, the data is going through the layers as in a pipeline. As depicted in Fig. 6, the information flows through the following layers: sensors, perceptual components, situation modelling, and services.

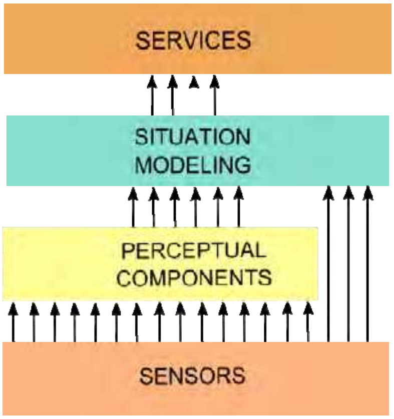


Fig. 6. Levels of abstraction in the smart environment architecture.

Perceptual components provide information about *entities*, which are mapped one-to-one to real world objects. These entities have a set of attributes, which may change over time. These attributes are modelled as *property streams*, with an update frequency, which depends on the type of attribute. For example, a person's ID will most likely change only a couple of times, depending on the accuracy of the person identifier technology, whereas the person's location will most likely be updated often.

The *situation modelling* layer is where the transformation of a set of facts about the environment into a set of situations is taking place (Crowley et al., 2002). The situation model will provide information about *situations*, which consist of a set of *states* and their associated *transitions*. The detection of the current state and the detection of state changes is implemented using several techniques, and is extendable. Currently, we have a rule-based engine, several statistical methods, and a very simple hardwired logic. Each situation is embodied in a *situation machine*, which is a Java module in SitCom, and only the external hull of the situation machine is visible. This uniformity of public interface for situation machines is a key aspect for the flexibility of the simulation environment. The actual set of used situation machines is application dependent.

The service layer contains smart environment's applications provided to the users or participants, examples of such applications are given in use cases in sections 7.2 and 7.3. More details about this architecture used herein can be found in (Dimakis et al., 2008).

6.3 The SitCom Tool

SitCom (Situation Composer for Smart Environments) is a 3D simulator tool and runtime for development of context-aware applications and services. Context-aware applications draw data from the surrounding environment (such as there is an ongoing meeting in the room, a person location, body posture, etc.) and their behaviour depends on the respective situation (e.g. while in meeting silent the phone). In SitCom, the environment characteristics are captured by Situations Models that receive input events from a) real sensor inputs (cameras, microphones, proximity sensors...), b) simulated data, and c) combination of real and simulated input. SitCom allows combining the situation models into hierarchies to provide event filtering, aggregation, induction and thus construct higher-level meaning and reasoning functionality. Through its IDE, SitCom allows to load, run or modify various scenarios with their subsequent realistic rendering in 3D virtual environment. These scenarios can be re-played to invoke situations relevant for the application behaviour, and thus provide mechanism for systematic testing of the context-aware application under different environmental conditions. The key feature is that we keep the portability between virtual and real devices.

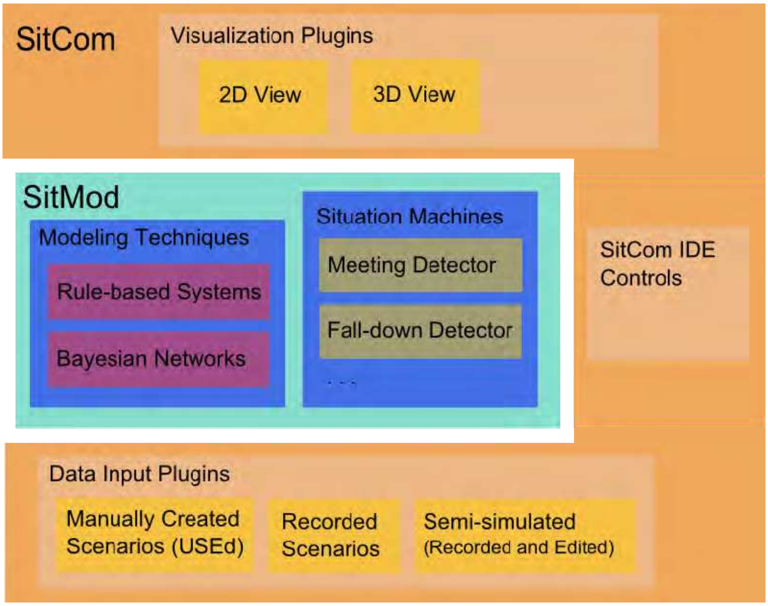


Fig. 7. Overview of SitCom's individual components and plugins.

6.3.1 SitCom's Components and Plugins

In the reference architecture model, SitCom itself plugs into the layer of situation modelling. So it sees the service layer as its upper layer, and the perceptual components as its lower layer. Also, as SitCom itself is a framework for modelling systems enabling plugging diverse situation modelling technologies, it is split into two distinct parts: the simulation framework, named SitCom, and the actual situation modelling being developed, named SitMod (Situation Modelling). In Fig. 7, the two parts are clearly separated. SitMod (and modules plugged into it) is the portion of the system to be deployed into the field. During simulation, but also once deployed, the SitMod run-time runs situation models for multiple and possibly unrelated scenarios. We found it very useful to be able to switch back and forth between simulated (manually created) and recorded data, as for debugging some situations, we can run the experiment up to a point, and then we can interfere with the recorded data using synthetic data, so as to test the situation model and the service. This makes the test of "what if" scenarios very quick and simple.

In the Fig. 8 there is main window of SitCom application. We can see visualization of the room in 3D virtual environment (Room-3D visualization plug-in) and in 2D visualization (Room-2D visualization plug-in). The visualization is generated based on data from input plug-ins. In the lower left part, there are controls for replay of the simulation and controls for handling of input plug-ins. On the right, there is video recording of the room, synchronised with both visualizations.

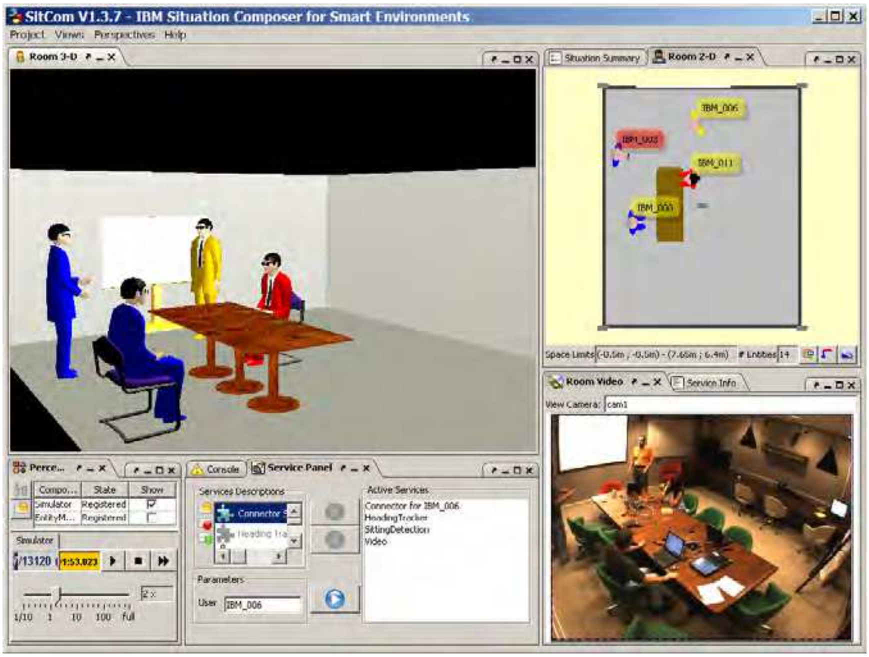


Fig. 8. Meeting scenario visualization using the SitCom tool.

6.4 USEd – User Scenario Editor

In contrast to SitCom, USEd is used to create, edit and tune user scenarios in ambient intelligence environment (especially in smart environments).

6.4.1 Main Principles

For the purpose of user scenario creation and editing task we developed the USEd, the **U**ser **S**cenario **Ed**itor. This tool supports most of the use cases presented in the task user scenario creation and editing, so it is possible to create single user scenario from external formats (e.g. audio/video recording), create single user scenario by usability expert, create cooperative scenario from external formats (e.g. audio/video recording).

The User Scenario Editor is an application for user scenario creation. A user scenario represents behaviour of the user in the ambient intelligence environment during usability evaluation. This editor is data compatible with SitCom, because it uses a SitCom environment description file for virtual environment description and one of the outputs of the editor is a SitCom person description file. Therefore scenarios recorded in the USEd can be investigated in SitCom runtime. Advantage of USEd is that it stores the user scenario as events instead of time snapshots of the environment status as SitCom. Therefore it is much easier to create and edit the scenario in USEd then in SitCom. USEd supports also simple verification of the created scenario so we can prepare the scenario without necessity to use SitCom to evaluate the scenario.

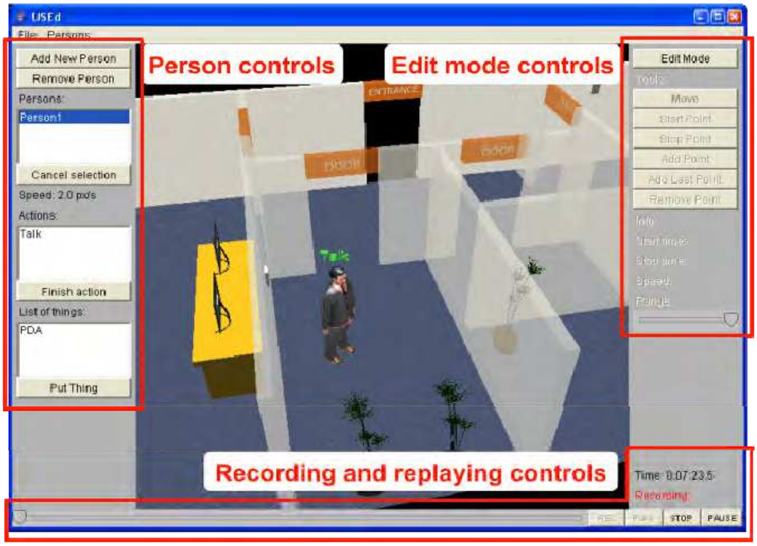


Fig. 9. Main window of USEd editor with.

The main screen of the USEd is divided into four parts, see Fig. 9. In the middle, the interactive virtual environment is displayed. USEd allows zooming, panning and rotation of the environment to allow full control during scenario editing. At the bottom, there are buttons for starting and stopping of the recording and for starting and stopping replaying of the recorded scenario. There is also the timeline showing time of the recorded scenario and allowing time shift during replaying. On the left, there are person controls for online recording. We can add or remove a person from the recorded scenario and we can select which person is controlled in case there is more than one person in the scene. There is also a list of actions the person is currently performing and list of things the person is currently holding. Two buttons are used for finishing the selected action or for putting the selected thing. On the right, there are edit mode controls with button toggling between data recording and editing.

6.4.2 Scenarios Creation and Editing

Scenario creation is the activity when a new scenario is created. It is started by pressing the REC button in bottom right corner. After that, timing begins and we are recording a new scenario. We add a new person by pressing the Add New Person button. A dialog for setting the name of the person appears and after entering the name the person appears in front of the starting point Entrance. Person movement is controlled by the clicking into the scene. After the clicking the person tries to move to the position where we clicked. Detection collision and path finding algorithms were implemented to simplify the scenario creation. Interaction of the person with other objects in the scene is performed by right mouse button clicking on the object. After that, a dialog menu with possible actions appears and we can select the desired action. There are two types of action; immediate actions and long lasting actions. The difference is that long lasting actions appear in the action list of the person and

must be explicitly finished compared to immediate actions that are executed immediately and do not appear in the action list of the person.
Editing is an activity suitable for correction of the recorded scenario. It is activated by pressing the Edit Mode button in the upper right corner of the application. In Edit mode the virtual environment changes (see Fig. 10), the person disappears and an individual person path is displayed instead. Also new buttons that switch between different user interaction modes are available. Edit mode supports movement of the navigation points, adding and removing points in the path and also adding new points to the path. Timeline controls the segment of the path that can be edited. This is extremely important in case the path is very dense in a particular area and helps us not to operate with wrong points of the path.

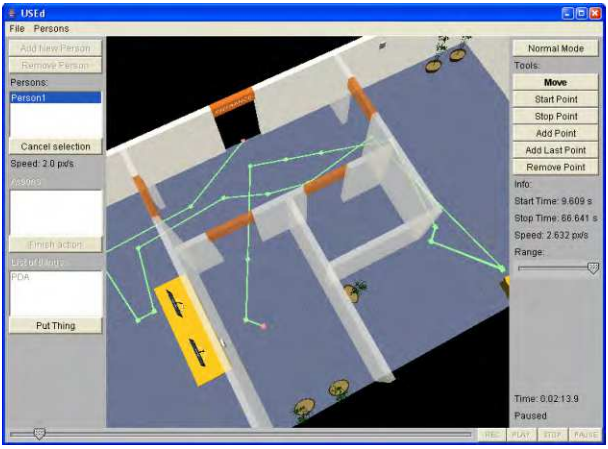


Fig. 10. USEd editor in data editing mode. Instead of avatar the movement path is displayed.

Collaborative scenario creation can be achieved by two different approaches. The first approach is one-by-one scenario recording, where one person is being recorded while other persons that were recorded are replayed to help with synchronization. The second approach is multiple persons recording, when all persons are recorded simultaneously. Using edit mode the actions are shifted to appear concurrently.

7. Use Cases

7.1 Inventory Use Case

The Inventory use case was used to evaluate the ability of the USEd editor to transform data from video recording into the virtual environment and create expert evaluation. Then, these two types of transformations were compared both from ease of creation and a quality of results point of view.
This use case was testing two parts of the framework, the task user scenario creation and editing and internal data storage.

7.1.1 Use Case Description

In this scenario the user is doing inventory in the smart environment with a PDA. The PDA has a barcode reader that allows for fast recognition of inventoried items. If the item is not in the correct place or in the correct room, the user can move the object in the application to the correct place. The scenario consists of the following steps:

- User enters the room.
- User looks around.
- User goes to the set of three LCD displays.
- User checks the displays and makes appropriate changes in the application using his/her PDA.
- User leaves the room.

7.1.2 Data Acquisition Setup

Expert evaluation data were created before the video of the real user was created. It is based on the scenario steps with estimated interactions and timings of user actions. The expert had no limit for data creation to avoid any influence.

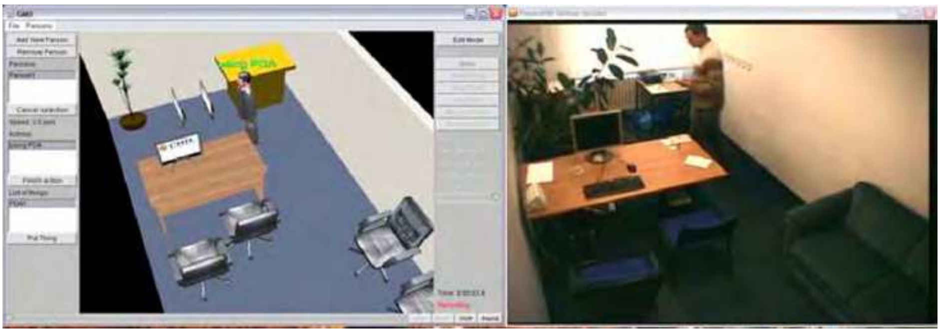


Fig. 11. USEd editor and video player during transformation of video into the virtual environment.

One video of the use case was recorded. Then 5 people tried to transform the video into the virtual environment using USEd. The computer setup is shown in Fig. 11. The USEd editor is on the left; the video player with a recorded scenario is on the right. Each person had to record the video in real time with an additional 5 minutes for corrections.

7.1.3 Use Case Results

All the people were able to transform the video into the virtual environment. There were some small usability issues of the USEd editor that will be corrected during future development. All the videos were similar, only a limited time drift of action start time appeared. Also, the algorithm for collision detection and path finding helped to unify the movement of the avatars.

7.2 Meeting in Smart Room Use Case

The meeting use case arose from actual needs in a project called CHIL (Computers in the Human Interaction Loop) (Waibel, 2009). Objectives of the CHIL project were to create a smart environment (in this case room) in which computers serve humans that focus on interacting with other humans. The use of expert-created scenarios at the beginning of the project allowed us to start development of services prior to setting-up all the required technologies. Later when the technologies were ready, it gives us the possibility to compare behaviour of the same service using either expert-created or real data. From the framework point of view, we were evaluating cooperative scenario, using real data and simulated data.

In our use-case example, we consider a connector scenario proposed and exploited in (Danninger et al., 2005). The connector service is responsible for detecting acceptable interruptions (phone call, SMS, targeted audio …) of a particular person in the smart room. During the meeting, for example, a member of the audience might be interrupted by a message during the presentation, whereas the service blocks any calls for the meeting presenter.

7.2.1 Use Case Description

The meeting state detector is implemented as a situation model in the SitCom framework (described in section 6.2). The situation modelling layer is the place where the situation context received from audio and video sensors is processed and modelled. The context information acquired by the components at this layer helps services to respond better to varying user activities and environmental changes. For example, the situation modelling answers questions such as: Is there a meeting going on in the smart room? Who is the person speaking at the whiteboard? Has this person been in the room before? Schemas in Fig. 12 correspond to the setup for the connector service scenario. In this case, the situation model provides the information about the current state of the meeting to the service.

7.2.2 Bootstrapping Situation Models by Scenarios Created in USEd

When a project is currently in its first stage, the “emerging” technologies planned to be used are usually under development, but service providers are already eager to try prototypes and discuss deployment of feature functions with possible clients. In such cases, the possibility to create a virtual space capable of simulating the technologies helps and speeds-up the whole development cycle of the project.

Schema in Fig. 12 shows the situation model created for a meeting state recognition task. We have investigated numerous experiments in order to find a suitable statistical method. According to the needs of CHIL services, the methods were applied to a task of detecting the state of the meeting using labels provided by the perceptual components. We use the facts produced by perceptual components (about people’s presence, location, pose, and voice activity in the room) to determine the meeting states. These facts may themselves be the result of the analysis of multiple sensors.

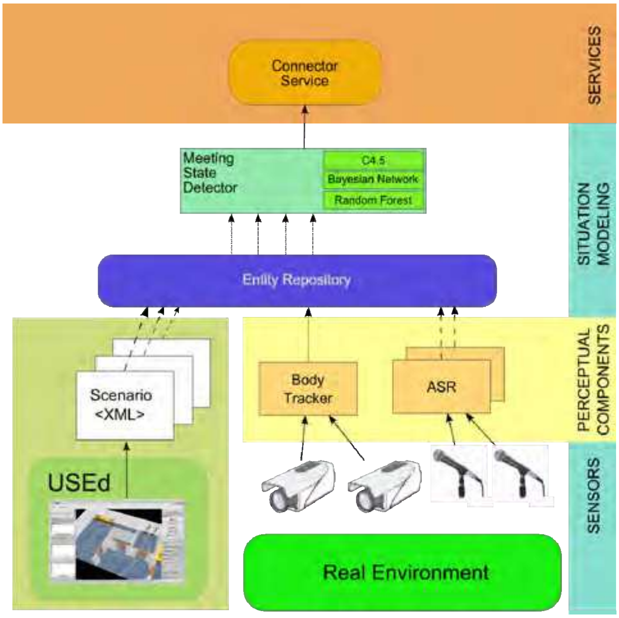


Fig. 12. Schema of data flow in SitCom for the Connector service use case.

7.2.3 Use Case Results

The expert-created scenarios (generated by the precedent of USEd) were used to bootstrap the initial work and to obtain first results. Identified methods and feature sets were then later used with seminar data, particularly the CHIL seminar 2007 corpus for both model training and evaluation. Fig. 8 shows visualization of recorded seminar data in the SitCom tool. A positive experience was that the feature sets and their parameters built and tuned on expert-created scenarios needed only minor changes when applied to real data. The differences were in selected time thresholds and intervals; the expert-created scenarios were not as dynamic as the real recording. Yet another minor difference was in expected and real speech activity among meeting participants; the expert-created scenarios contained many more interruptions and noisy conversations than the real case. We have used five manually crated scenarios of total length 1 hour 25 minutes; the real recordings comprised 6 scenarios of 1 hour 50 minutes length.

7.3 Home for Elderly Use Case

This use case is the most complex evaluation of our framework. First of all, we are testing detection mechanism accuracy and then we are evaluating single user scenario using both real and simulated data.

7.3.1 Use Case Description

Another more recent use case based on the approach advocated in this paper is Netcarity project. Netcarity is an ongoing integrated project supported by the European Community under the Sixth Framework Programme (IST-2006-045508). The aim of this project is investigation and testing of technologies which will help elderly people to improve their well being, independence, safety and health at home.

Fig. 13. SitCom's 3D visualization of a smart home.

The idea of this project is to equip homes of Netcarity clients by a variable set of perceptual technologies and connect them to the central point providing various types of services. The most important activity of the central service point, called Netcarity Server (NCS), is to "keep an eye" on the client health and safety exploiting different perceptual technologies, i.e. acoustic event detector, body tracker.

This project is currently in its first stage when the technologies are still under development, but service providers are already eager to try prototypes and discuss deployment of feature functions with possible clients. We have created a SitCom virtual smart home and several scenarios simulating essential situations in inhabitants' life, see Fig. 13. The following sensors, actuators and perceptual components are being simulated in the virtual home environment:

- Motion Detectors are installed in each room, monitoring movement in the space.
- Fire Detector sensor detecting fire in the flat.
- Gas Detector sensor detecting leaking gas in the flat.
- Remote Door Lock sensor/actuator capable of reporting open/closed status of the door and locking/unlocking the door's lock.
- Scene Acoustic Analyzer analyzing noise and sounds in the scene, such as, someone is walking, speaking, laughing, door is opening/closing.
- Body Tracker is a video-based tracker providing 2D coordinates for each person in the room.

- Talking Head actuator giving audio/visual feedback to the inhabitant. It serves as a communication channel from the Netcarity server and as a humanoid interface to other services.

The example of situation machine is:

- Fall-down Detector perceptual component detecting that the inhabitant may be in dangerous situation, for example he/she felt down and is unable to call for help. It uses and combines information from different sensors and detectors.

7.3.2 Bootstrapping by Scenarios Created in USEd

In the actual implementation the sensors and actuator in client home are controlled by a Residential Gateway (RGW) which is connected to the Netcarity Server by a secure and authenticated channel, called Netcarity Secure Channel (NCSC). In the simulated case, the residential gateway is replaced by the SitCom run-time connected to Netcarity Server via the NCSC connection. There is no difference in behaviour or data types between simulated and real-live smart home from the Netcarity Server side. Multiple instances of SitCom run-times can simulate a whole network of smart homes and their residential gateways. This setup is currently used for tuning Netcarity Server's rule-based system, fall-detector and acoustic scene analysis modules, and for verification of proposed communication schemes. Beside the visual feedback during the development, the 3D visualization obviously serves also as a good demonstration platform.

7.3.3 Use Case Results

We have used USEd to create several artificial scenarios to test the behaviour of the whole Netcarity setup. Using simulated data to support fall-down scenario was crucial, as it is unlikely to record real fall in the real home. We have used semi-simulated approach in that case, i.e. using real recordings of human movements and actions combined with simulated falls. In addition, we have recordings of various activities of daily living and "forced" falls from lab environment (Libal et al., 2009).

8. Conclusions and Future Work

In this work, we introduced framework for visual analysis of user behaviour in ambient intelligence environment. The framework consists of five tasks that are performed during evaluation of the applications, depending on the use case. These tasks are divided into two groups. Testing of detection mechanism accuracy task and user scenario creation and editing task are tasks that belong to first group of tasks that focus on preparation of data for tasks in second group. Second group tasks focuses on visual analysis of user behaviour either in evaluation of single user scenarios task or in evaluation of cooperative scenarios task. The framework tasks are executed using two tools, SitCom and USEd. These tools have been exercised by many developers over the course of several years. The tooling framework is currently used by more than ten sites across Europe and it has been part of several technology demonstrations. To support the development within the Reference Architecture Model, we have implemented SitCom framework – an extensible Java toolkit to help in all phases of the development of the non-trivial life-cycle of context-aware services. We equipped SitCom with a set of functionalities that we found beneficial in development of perceptual applications:

- it simulates the environment and the perceptual input (manually created or recorded scenarios)
- it provides 2D and 3D visualizations of scenes, situations, and scenarios
- it works as a middleware between user services and the layer of perception (situation modelling)
- it supports the context model builders with a framework for data fusion and filtering
- it serves as an IDE for repetitive testing of context-aware services and applications by replaying recorded or simulated scenarios
- it provides portability between virtual and real devices

The SitCom environment was successfully applied during the whole development cycle of the CHIL platform and it is now helping to bootstrap the Netcarity system. In many ways, having an integrated tool specialized for context-aware applications has been helpful to identify the necessary pieces for the application, like the needed room information, specific sensor information, or the definition of the participants roles.
The USEd tool has been used to create, edit and tune user scenarios in ambient intelligence environment (especially in smart room). The tool proved to be not only usable add-on to the SitCom in cases of simulated scenarios creation and editing but also fast tool for transformation of video recordings into the user scenarios.

Usage of the framework and applications was demonstrated in three use cases, where a subset of tasks was used. Results of the use cases showed that both tools are usable and they are able to work with data and visualize them.

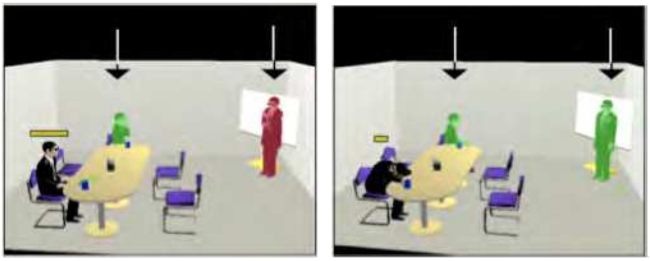


Fig. 14. Visualization of cognitive load of the user (black sitting person).

In the future work, we would like to focus on improvement of visualization in order to increase value of the visual analysis in evaluation of the ambient intelligence application. Currently, the visualization and data collection focus on behaviour of the user, like her/his position, her/his movement, head direction or detection of speech. We would like to add more information about the user interaction with objects in the environment. Such a type of visualization is closely linked up with the user model. This means that the advantage of this type of visualization is, that we can directly estimate the user cognitive load from the available data (user interaction log files). The cognitive load is rising, when the user is doing interactions, which are not expected by the developer, or the user is not sure how to continue to finish her/his task, and so on. By means of this visualization tool the evaluator gets instantly the global overview of the user performance (together with idea about critical points in the user interface investigated). The potential tool that can visualize behaviour of users (avatars) might generate the output given in Fig. 14. In such a way it is possible to investigate detail behaviour of individual users in complex situations. By means of colours it is possible to mark visually some specific features (parameters) of avatars that occurred in specific situations (in this case the cognitive load of the user in particular situations).

9. Acknowledgements

We would like to acknowledge support of this work by the European Commission under IST FP6 integrated project Netcarity, contract number IST-2006-045508. This work has also been partly supported by the Ministry of Education, Youth and Sports of the Czech Republic under the research program LC-06008 (Center for Computer Graphics) and the research program 6840770014 (Research in the Area of the Prospective Information and Navigation Technologies).

10. References

Barton, J. J. & Vijayaraghavan, V. (2002). UBIWISE, *A Ubiquitous Wireless Infrastructure Simulation Environment*. HP Labs Technical Report. HPL-2002-303.

Crowley, J. L.; Coutaz, J.; Rey, G. & Reignier. P. (2002). Perceptual components for context aware computing. In *Proceedings of UBICOMP 2002, International Conference on Ubiquitous Computing*, Goteborg, Sweden, pp. 117-134, September 2002.

Danninger, M.; Flaherty, G.; Bernardin, K.; Ekenel, H.; Köhler, T.; Malkin, R.; Stiefelhagen, R. & Waibel, A. (2005): The connector: facilitating context-aware communication. In: *7th Int. Conference on Multimodal Interfaces 2005*. ICMI'05, New York, USA, ACM Press. 2005.

Dey, A.; Salber, D. & Abowd. G. (2001). A conceptual framework and a toolkit for supporting the rapid prototyping of context-aware applications. *Human-Computer Interaction (HCI) Journal*, Vol. 16, pp. 97-166, 2001.

Dimakis, N.; Soldatos, J. K.; Polymenakos, L.; Fleury, P.; Cuřín, J. & Kleindienst, J. (2008). Integrated Development of Context-Aware Applications in Smart Spaces, *IEEE Pervasive Computing*, vol. 7, no. 4, pp. 71-79, Oct-Dec, 2008

Fleury, P.; Cuřín, J. & Kleindienst. J. (2007). SitCom - development platform for multimodal perceptual services. In *Proceedings of 3nd International Conference on Industrial Applications of Holonic and Multi-Agent Systems*, Regensburg, Germany, September 2007. V. Marik, V. Vyatkin, A.W. Colombo (Eds.): HoloMAS 2007, LNAI 4659, Springer-Verlag, pp. 104-113, 2007.

Gallud, J. A.; Lozano, M.; Tesoriero, R. & Penichet, V. M. R. (2007). Using Mobile Devices to Improve the Interactive Experience of Visitors in Art Museums. *In Human-Computer Interaction. Interaction Platforms and Techniques*, volume 4551 of LNCS, pp. 280–287. Springer, 2007.

Harrison, B. L.; Owen, R. & Baecker, R.M. (1994). Timelines: An Interactive System for the Collection of Visualization of Temporal Data. *In Proc. of Graphics Interface'94.* Toronto: Canadian lnfo. Proc. Society, pp. 141-148, 1994.

Ivory, M. Y. & Hearst, M. A. (2001). The state of the art in automating usability evaluation of user interfaces. *ACM Computer Survey*, Vol. 33(4), pp. 470-516, 2001.

i2home. The Intuitive Interaction for Everyone with Home Appliances based on Industry Standards (i2home) project. Web site (2009) http://www.i2home.org

Libal V.; Ramabhadran B.; Mana N.; Pianesi F.; Chippendale F.; Lanz O. & Potamianos G. (2009). Multimodal Classification of Activities of Daily Living in Smart Homes, *International Workshop of Ambient Assisted Living (IWANN 2009)*, S. Omatu et al. (Eds.), LNCS 5518, Springer, pp. 687-694, 2009.

Maly, I. & Slavik, P. (2007. Towards Visual Analysis of Usability Test Logs Using Task Models. In *Proc. of the 5th Workshop on Task Models and Diagrams 2006*. TAMODIA 2006. LNCS, vol. 4385, pp. 24–38. Springer, Heidelberg, 2007.

Mikovec, Z.; Maly, I.; Slavik, P. & Curin, J. (2007). Visualization of users' activities in a specific environment. In *Winter Simulation Conference (WSC) 2007*, pp. 738–746, Washington DC, USA, December 2007. WSC.

Nazari Shirehjini, A. A. & Klar, F. (2005). 3DSim: Rapid Prototyping Ambient Intelligence. In *Proceedings of the 2005 joint conference on Smart objects and ambient intelligence: innovative context-aware services: usages and technologies*. ACM International Conference Proceeding Series; Vol. 121, pp. 303-306, 2005.

Nazari Shirehjini, A. A. (2005). A Generic UPnP Architecture for Ambient Intelligence Meeting Rooms and a Control Point allowing for integrated 2D and 3D Interaction. In *Proceedings of the 2005 joint conference on Smart objects and ambient intelligence: innovative context-aware services: usages and technologies*. ACM International Conference Proceeding Series; Vol. 121, pp 207-212, 2005.

Nielsen, J. (1994). *Usability Engineering*. Morgan Kaufmann.

Nishikawa, H.; Yamamoto, S.; Tamai, M.; Nishigaki, K.; Kitani, T.; Shibata, N.; Yasumoto, K. & Ito, M. (2006). Ubireal: Realistic smartspace simulator for systematic testing. In *Ubicomp*, volume 4206 of LNCS, pp. 459–476. Springer, 2006.

Paterno, F.; Mancini, C. & Meniconi, S. (1997). Concurtasktrees: Diagrammatic notation for specifying task models. In *IFIP TC13 Sixth International Conference on Human-Computer Interaction*. pp. 362–369, Chapman and Hall, July 1997.

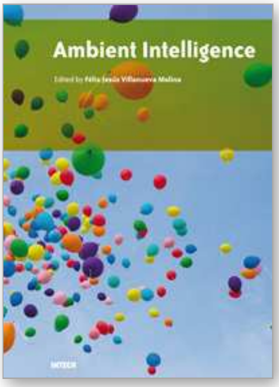
Paterno, F.; Russino, A.; Santoro & C. (2007). Remote Evaluation of Mobile Applications. In *Proc. of the 6th Workshop on Task Models and Diagrams 2007*. TAMODIA 2007. LNCS, vol. 4849, pp. 155–169. Springer, Heidelberg. 2007.

Rubin, J. (1994). Handbook of usability testing: *How to plan, design, and conduct effective tests*. John Wiley & Sons, Inc., New York, NY, USA, 1994.

Seo, J.; Goh, G. & Kim, G. J. (2005). Creating Ubiquitous Computing Simulators using P-VoT. In *Proceedings of the 4th international conference on Mobile and ubiquitous multimedia*, ACM International Conference Proceeding Series; Vol. 154, pp. 123-126, 2005.

Spence, R. (2000). *Information visualization*, ACM Press, Addison Wesley, 2000

Waibel, A. & Stiefelhagen, R. 2009. Computers in the Human Interaction Loop. *Human-Computer Interaction Series*, 2009, XXII, 374 p. 108 illus., 88 in color., Hardcover, Springer, Heidelberg, 2009.



ISBN 978-953-307-078-0

Hard cover, 144 pages

Publisher InTech

Published online 01, March, 2010

Published in print edition March, 2010