**Noname manuscript No.**
(will be inserted by the editor)

# A Mobile 3D-User Interface for Mixed-Initiative-Interaction in Ambient Intelligence Environments

**Ali Asghar Nazari Shirehjini** ·
**Azin Semsar**

the date of receipt and acceptance should be inserted later

**Abstract** This paper describes concepts, design, implementation, and performance evaluation of a 3D-based user interface for accessing Ambient Intelligence Environments (AmIE). The generic interaction model of the described work addresses some major challenges of Human-Environment-Interaction such as cognitive overload and manual device selection, loss of user control, missing system image or over-automation. 3D visualization and 3D UI, acting as the central feature of the system, create a logical link between physical devices and their virtual representation on the end user's mobile devices. By so doing, the user can easily identify a device within the environment based on its position, orientation, and form, and access the identified devices through the 3D interface for direct manipulation within the scene.

**Keywords** Mobile 3D UIs · case studies of 3D UIs in real-world use · applications of 3D UI techniques · Ambient Intelligence

## 1 Introduction

Ambient Intelligence (AmI) has the vision of pervading our everyday environment and its objects by sensing, computing and communication capabilities [13]. A major characteristic of such environments is the increasing amount of intelligent devices and their complexity. Computing infrastructure will be ubiquitous and make accessible things like electricity, hydro or telephone everywhere and anytime. However, due the weaknesses of existing user interfaces, when interacting with Ambient Intelligence Environments (AmIE), users are overloaded because of the

A. A. N. Shirehjini
Computer Engineering Department, Sharif University of Technology, Tehran, Iran
E-mail: shirehjini@sharif.edu

A. Semsar
Computer Engineering Department, Sharif University of Technology, Tehran, Iran

multitude of networked appliances and their increasing functional complexity [17]. Identifying and activating the right device or service from a huge amount of existing devices to perform a specific task is a challenge for users of these environments. As devices grow in complexity and sophistication, they become harder and harder to use, particularly because of their tiny display screens and limited interfaces. At the same time devices will disappear into the background and will be invisible to the user. However, because of this transparency users fail to develop an adequate mental model of the AmIE and its interaction concept [36]. The reasons for this are the seamless integration of the infrastructure into the background and the missing or invisible user interfaces. To overcome these challenges, new interaction models are required. How can one interact with tiny devices that do not provide their own user interfaces? Or how to find and access devices in an environment which are invisible to the user? How to access physical devices in an unfamiliar environment without having knowledge about the technical infrastructure such as device's physical address or IP address? In this article, we present a mobile 3D user interface system that facilitates user interaction in Ambient Intelligence Environments. We will outline challenges which are common in such environments, and propose a solution to overcome them. We also present a proof of concept implementation and a user evaluation. But first, we start with an overview of related work and existing approaches.

## 2 Related Work

The work presented in [14] and [27] sketch the basic ideas of Ambient Intelligence and intelligent environments and the characteristics of AmIE. F. Sadri [35] and Kashyap et al. [25] give an extensive overview about challenges of Ambient Intelligence. Gilman et al. [21] and Poslad [32] provide a good overview of existing interaction systems. The investigations in these have lead to the questions and solutions of our work.

A significant number of recent research have focused on interacting with next generation smart homes and how users will interact with them [15, 16, 18]. Past research [25] have underlined the problems of over-automation, loss of control, complexity of the environment, and lack of appropriate user interfaces for smart everyday devices. A study conducted by Allouch et al. [15] indicates that a full-automated system is not accepted by users; instead a complementary explicit interaction should be available. However, these challenges have remained under-explored. Although few ambient interaction systems have been proposed in the past, the topic still needs further research.

[22] proposes an Augmented Reality (AR) based interaction concept to overcome above-mentioned problems. But the approach is not appropriate for large environments because it only recognizes devices which are within the capturing zone of the camera. In contrast, we provide a 3D based user interface which overcomes aforementioned interaction challenges, and allows the user to explore larger environments using the 3D-scene. The 'camera view'can be automatically moved to a certain part of the 3D-scene depending on the current user situation, i.e.

user position and orientation, current activity and environment context. This is not possible with AR. Our system is also much easier to use and inexpensive, as it avoids the use of Head Mounted Display (HMD) that is necessary in an AR solution.

User input for manipulating devices or virtual objects via AR-based systems is still a challenging task. In contrast, multi-touch input on touch sensitive mobile phones can easily be used for 3D object manipulation.

Some researchers [31, 42] have proposed using natural language for interacting with smart spaces. Generally, this approach can be used to identify and select the right device without requiring knowledge of technical infrastructure (e.g. device URLs). However, speech interaction is not appropriate for many domains. For example, when presenting a PhD thesis, the candidate might not want to speak to the projector. The speaker's speech and gesture modalities are already busy because they are used for Human-Human-Interaction. Moreover, robust speech recognition in public spaces for multiple users is another challenge of speech and gesture interaction.

Finally, none of the existing work discusses how to allow an explicit interaction system to coexist with an implicit adaptive environment.

In contrast, we introduce a novel interaction model based on combining explicit and implicit interaction components. Our explicit system is based on a 3D user interface that overcomes the problems of intuitive device matching in complex environments. The 3D user interface allows users easy exploration of the AmIE. Interactable devices can be found and understood [28] easier, supporting the development of adequate mental models [soltani,saniee]. To the best of our knowledge, no other work allows for mixed-initiative interaction for Ambient Intelligence.

## 3 Challenges in Human-Environment-Interaction

In general, Human-Environment-Interaction is divided along the dimension of the initiative in two classes [29]: explicit and implicit interaction. Within the scope of an explicit interaction, the user takes the initiative to control the environment and decides when to perform which activity. In contrast, by an implicit interaction actions are performed automatically and the environment takes the initiative to perform activities. The user does not determine directly when something happens. Instead, users are observed by the system to understand their current behavior and situations. This lack of control and overautomation is a weakness because people do not accept a full-adaptive and “over-automated” environment [24], and in fact want to always be in control [12].

Another challenge of implicit interaction is the lack of system interfaces [19, 20]. This makes it difficult for users to build an appropriate mental model about the system [26] and to understand the automated (re)actions of their intelligent environment.

One component in the successful use of automated systems is the extent to which people trust the automation. In order to constitute trust, users need to understand the current system's state (‘visibility’) and to predict its behavior [23, 24]. Thus, an adequate representation of the system is required to support the development of user's mental models as well as their trust in the automated system.

Yet another interaction challenge is the inability of users to override the default behavior of the system. Explicit interaction bypasses the challenges of implicit interaction, but has its own issues: One is the difficulty of designing an intuitive device selection interface. Users are typically confronted with the full scale of the environments' complexity and can become distracted from their real tasks. This becomes even more difficult in an unknown environment. In most of the existing user interfaces, target devices have to be identified and selected based on 2D-icons, complex menus, or device numbers. It is truly a big challenge to find and activate needed devices [20], especially in foreign and complex environments. Even natural gesture-based interactions do not solve this problem since one cannot easily point to a device which is seamlessly integrated into the environment.

In the next chapter, we introduce a novel interaction approach, to overcome the discussed problems. The interaction model is based on the idea of mixed-initiative interaction and uses a mobile 3D user interface to provide explicit interaction. To handle possible interaction conflicts, the system provides mechanisms for conflict management.

## 4 Proposed Solution

To address the above challenges, we follow a mixed-initiative interaction paradigm to overcome the weaknesses of both of the said interaction paradigms. In order to solve the problem of manual device selection, we suggest to use 3D-based user interfaces to allow the end user to identify and select devices based on their spatial relationships without requiring technical information such as device IP addresses. This lessens the cognitive effort for exploring and accessing complex and dynamic environments.

Our interaction model integrates document access and device access in the same user interface, allowing drag&dropping a document to a physical device within one single interaction step. The Abstract Interaction Model in Figure 1 shows the overall interaction model. It combines the benefits of implicit and explicit interaction, and on one hand, allows the user to stay in the control loop because there is always the possibility to intervene via the explicit interaction system, and, on the other hand, supports the user by the implicit system. Thus the benefits of automation will still remain while eliminating the problem of over-automation as the user will be able to decide which activities are permitted for an implicit interaction. Due the concurrent nature of this mixed-initiative approach, interaction conflicts can exist. To avoid conflicts such as opposed actions and inappropriate automatism of the adaptive environment, the implicit and explicit interaction will be coordinated and synchronized. For example, in a thesis presentation scenario in a smart lecture room, the environment could provide some adaptivity such as reactive interaction with air conditioning and lighting devices while the user could use the explicit remote control assistants to explicitly manage the presentation. Furthermore, the user can regulate the level of automation and correct wrong automatisms or inappropriate adaptation of the environment in case it happens, leading to a higher level of trust [24].

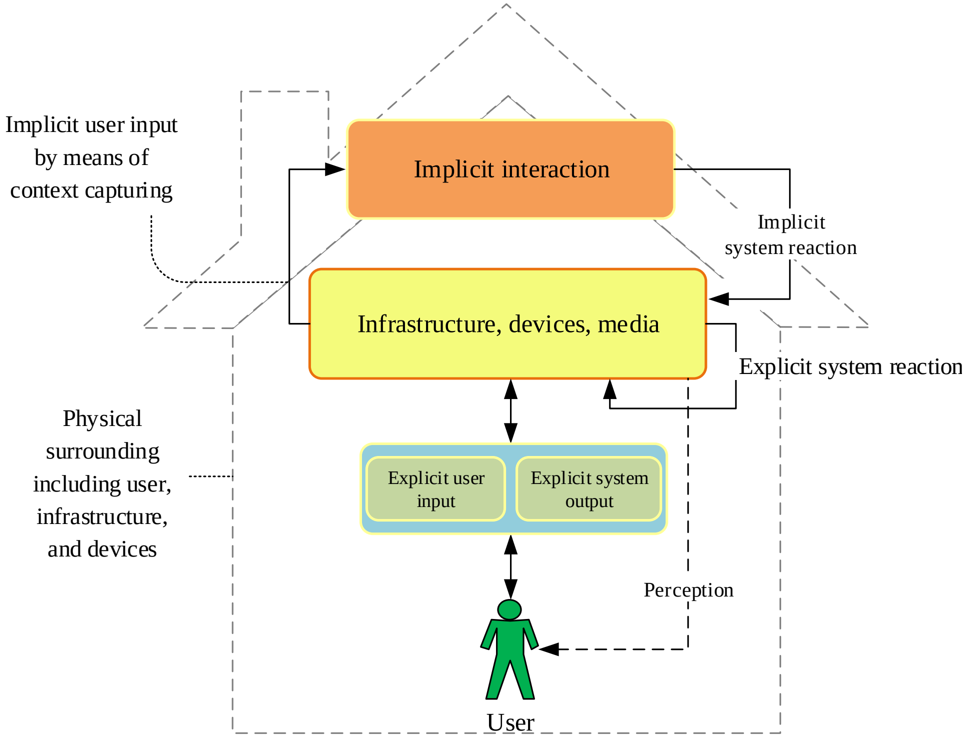


**Fig. 1:** *Abstract interaction model*

The interaction model facilitates the room metaphor (cf. Fig. 2), allowing users to select devices as they do it in a real world. Users can identify and select devices based on their position, orientation and form. Thus, device selection happens without requiring technical knowledge such as IP addresses for identification purposes. Instead, the spatial topology of devices is used for mapping UI controls to the physical devices.

Several types of input forms (i.e. modality and user input devices) could be imagined. Hand gestures and pointing devices both would allow to select a device in a room by simply pointing at it. However, we believe using a 3D user interface has several advantages. One reason is that users will have difficulties to point to some tiny devices in bigger rooms. Another problem of hand gestures and pointing devices is that users can not select invisible devices using these metaphors. In contrast, a 3D user interface can visualize the environment and its devices and create a direct correspondence between the physical objects and their 3D representations within the user interface. Device and device states can then be visualized within a 3D interface. Even devices that are hidden to users but are of interest at a given situation can be visualized. Device states and recent actions of devices such as moving a shutter could be animated to the user in order to create awareness for

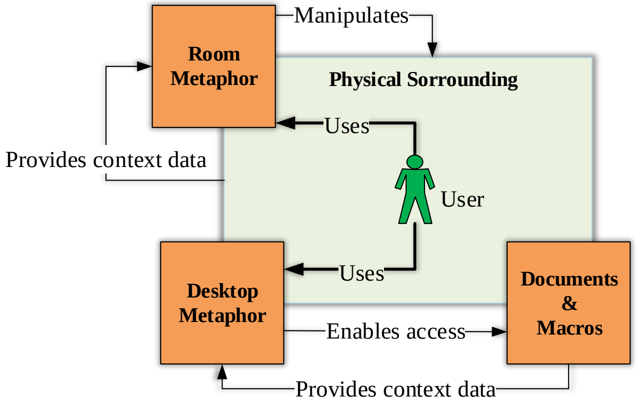


**Fig. 2:** *Combination of desktop and room metaphor*

changes. By so doing, the environment becomes more visible for the user and help the user better orient himself within the environment.

Moreover, the visualization of the environment provides a system image in the AmIE which helps the user build an adequate mental model for interaction, leading to less cognitive effort for exploring and accessing complex and dynamic AmIE.

As another benefit, the described interaction model integrates the user's virtual world and its objects (i.e., files) with his physical world by providing an integrated user interface and allowing the user to directly manipulate these two worlds simultaneously. By combining the room metaphor [30] and desktop metaphor [43], we provide users with a single point of access to their documents and devices. For instance, the user can move a presentation file from a notebook to the projector and get it started there with only one drag-and-drop operation. By doing so, we extend the well known and accepted interaction paradigm of direct manipulation [41] from the virtual computer world to the physical world.

As mentioned before, the user can also avoid, stop, or undo inappropriate implicit actions and adaptations, which provides a basic conflict management. Allowing the user to remain in control and increasing the reliability of the AmIE will increase trust in the automated component of the system thus motivating the user to accept and use of the technology. The solution proposed here was implemented and tested in a real scenario, as described next.

## 4.1 The Integrated User Interface

At an abstract level, the user interface is composed of the following entities (cf. Fig. 3 and 4):

- Document view provides a single-point of access to personal media;

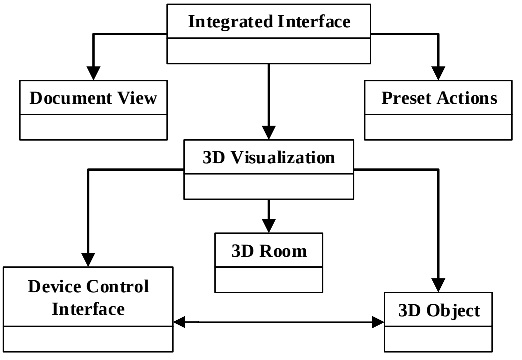


**Fig. 3:** *The abstract presentation model shows the entities of the UI*

- 3D Visualization provides a system face to the physical environment by visualizing devices and their states;
- Preset Actions allow the user to store a family of related actions as a macro and to use it within a certain environment.

We implemented the Ambient Interaction System (AmIS): a novel interaction system that implements the abstract interaction model described in the previous chapter. It is a personal assistant system for environmental control. AmIS integrates the virtual media repository as well as the physical environment of the user into a unified personal environment (cf. Fig 4 and 5). AmIS uses an automatically created 3D visualization model of the environment. Entering a room, it discovers the infrastructure and available devices and builds the integrated user interface. Changes to the environment, new devices, or re-positioned devices are identified and implicate an update of the UI. Henceforth, the user can access identified devices through the 3D interface and directly manipulate them. For example, the user can just tap on a 3D object representing a specific bulb in the room to turn on the light. AmIS also contains a Windows Icons Menus and Pointing (WIMP) based 2D interface to provide access to documents such as slides, pictures, music, and video. The user can drag&drop objects representing personal media to devices in the physical environment as well as to pre-set actions. At the bottom left of the interface (cf. Fig. 4), there is a view of the user's personal data. This view provides the user access to all ubiquitous data, which can be distributed among several devices. The access to ubiquitous data helps the user to fetch quickly movies, music, presentations (PPT files), documents, and pictures regardless of their physical storage location. Another feature of the interface is the list of preset actions at the bottom right of its screen. Using AmIS, the user can select a movie, drag it from the Personal Media repository (shown and accessible by the integrated UI), and drop it into a specific TV or display in the 3D view of the physical environment. This action starts the movie of the chosen file on the selected TV. In this example, the user associates media with an object in the 3D view by a single interaction step. Within

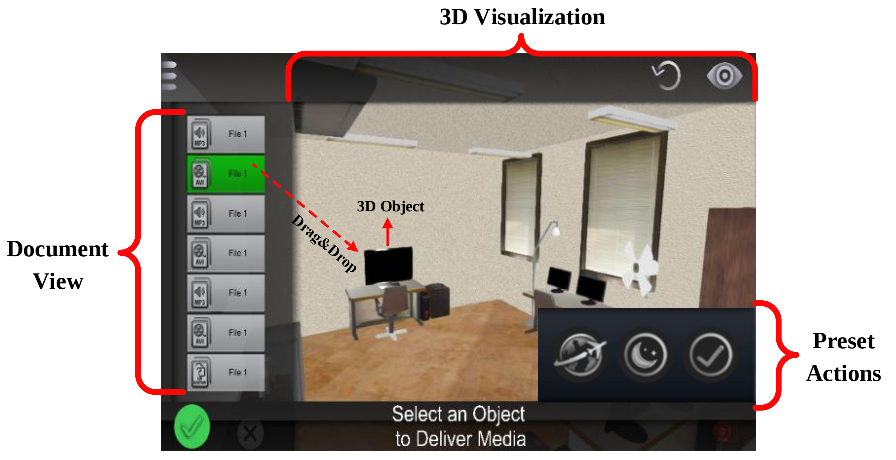


**Fig. 4:** *The UI of the AmIS system has three parts: the 3D view, the document view and the preset actions.*

the 3D-interface, the user can also control several lights, jalousies and blinds, HVAC devices, or even the speakers of the room by simple tapping on an object. For example, in a large living room, turning on the third light from the left is as easy as tapping on the third light from the left in the 3D interface.

AmIS also allows advanced actions to be done by using specific, pre-compiled user interface widgets which we call Device Control Interfaces. A Device Control Interface pops up when a device providing such an interface has been tapped. For both 3D and 2D interaction, AmIS provides unified metaphors (pointing, selecting, drag&drop, tapping) so the user can handle displays and lights in the same manner as interacting with files and directories. By so doing, we extend the well-known metaphor of direct manipulation to the physical world.

### 4.2 AmIS Architecture

The AmIS system has a distributed architecture as shown in the Figure 6. The Interaction Appliance (IA) runs on a mobile phone making available the AmIS client and the user interface. The Environment Manager (EnvMan) runs in the cloud. The Dynamic User Interface Creator (DUIC) is an agent software which is in charge of the dynamic 3D-Visualization and has been implemented as a built-in component of the the EnvMan. The 3D Objects database is a web-server running on the cloud. Finally, some parts of the AmIS systems will run directly on the devices of the environment, including UPnP stack and UPnP agents which allow invocation of device actions and also distribution of device state events. Despite not being

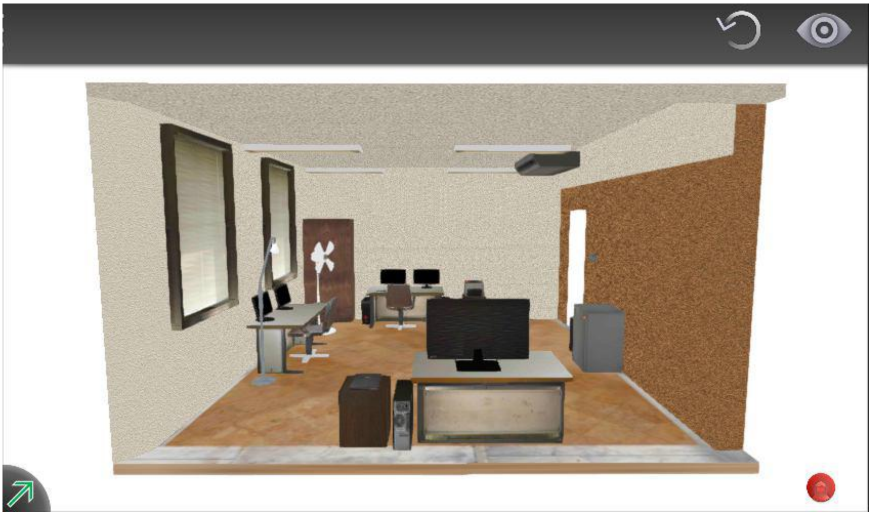

**Fig. 5:** *Screenshot of the UI of the AmIS system showing devices in the environment in the physical environment of the user*

pictured, the IoT Gateway plays a crucial role in the system that translates messages between the system and the devices that doesn't understand UPnP or DLNA [3, 8]. It is also capable of using several types of physical links (e.g., IEEE 802.3 [11], IEEE 802.11 [10], RS 232 [34]) to connect to Smart TV and other devices that expose a serial, infrared, or IP based interface. Moreover, using higher level software and network protocols, the IoT Gateway bridges various devices and sensor networks such as the ZWAVE Plus [9], Ant+ [2], PlugWise [6], KnX [5], Allnet [1]. This allows AmIS to integrate a broad range of networked devices such as multimedia home equipment, kitchen appliances, building management systems (BMS), biophysiological health monitoring gadgets, environmental sensors, and power consumption measurement devices. Although the establishment and management of the necessary physical links are done by IoT Gateways, retrieved sensory data are handled by another main component of AmIS, the Environment Monitoring (EnvMon). It is responsible for device discovery and monitoring. This part of AmIS observes the whole environment and provides information about important changes to the other components such as the Interaction Appliance and automation agents. These situation updates can for instance include a newly found device, a change in a device property like the on/off state of a device, or a change in room properties like the room temperature. To achieve this, several sensors and sensor systems are monitored to collect and aggregate the provided context information. The EnvMon component runs on the cloud. For an alternative local hosting server-side servers, various machines are available, including powerful single-board computers such as the Rasperi Pi 2 [7] and setup boxes such as the AVM's Fritzbox 7490 [4]. The EnvMon component is accessible for other system entities in terms of KQML based

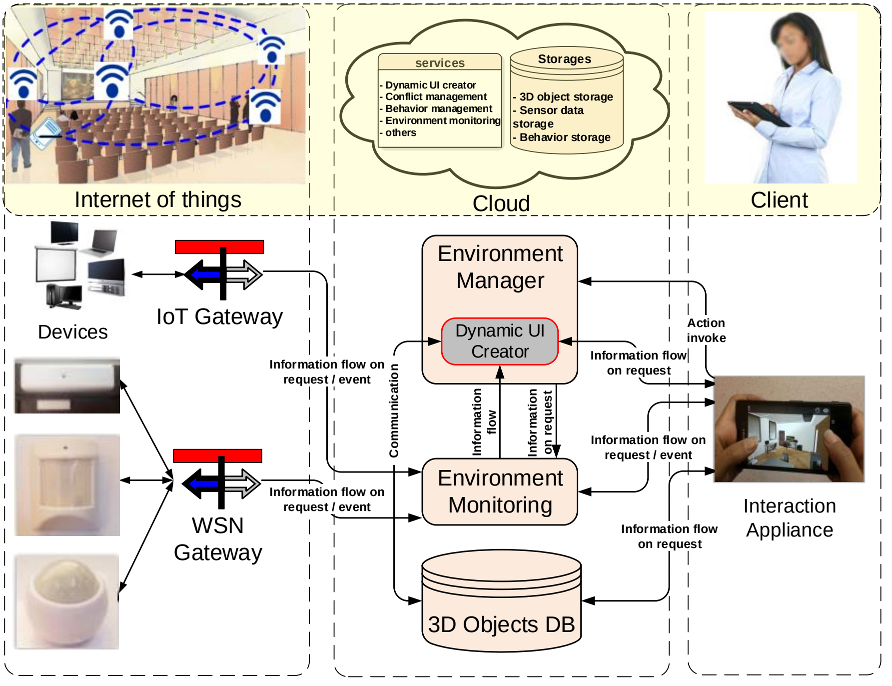


**Fig. 6:** *The system runs on several distributed machines and subsystems*

communication. Currently, the context information management is realized by an OWL based ontology, the details of which is beyond the scope of this paper [44].

## 4.3 The Environment Manager

The Environment Manager (EnvMan) is the central part of the AmIS system. The overall responsibility of the Environment Manager is to mediate between the user side and the room side (cf. Fig. 6). Towards this goal, the EnvMan provides the following functionalities:

- **Devices Access**: the main task of the EnvMan is to invoke functions on devices and to perform some media management tasks upon user request. For this, EnvMan uses IoT Gateways to access heterogeneous devices and media repositories of various protocols and standards.
- **Dynamic User Interface Creation:** the EnvMan creates an XML file describing the current state of the environment which is to be represented by the 3D user interface of the interaction appliance.
- **Context Retrieval:** in order to create the above mentioned XML file, the EnvMan uses the Environment Monitoring (EnvMon) component to retrieve

context information about the room, available devices and their states as well as geographical information about device and users.
- **Interaction Synchronization:** the EnvMan is also in charge of managing simultaneous access to the environment as well as the coordination of concurrent interactions. By concurrent interaction we mean the coexistence of implicit and explicit interaction agents as well as multi-user interaction in terms of several AmIS users accessing the environment simultaneously. The interaction synchronization is a subsystem of the EnvMan which uses concurrency control mechanisms in order to manage coexisting interaction units as well as simultaneous access to the environment.

## 4.4 Information Flow

The internal communication in AmIS is done on the basis of a self-defined communication protocol that uses one standard network protocol as carrier. At the moment, KQML is used to transport AmIS messages as payload between single components. Outside the system, IoT gateway using device specific protocols such as UPnP, ZWAVE, etc. are used to invoke actions on devices. At the startup, the Environment Manager needs a complete environment overview. Details about the room's 3D geometry and properties and available devices is necessary for handling and checking the incoming commands. Normally the Interaction Appliance (IA) sends commands to the Environment Manager in order to change the environment, but before sending anything to the Environment Manager, the IAs subscribe to the system to indicate they are currently active. Similarly, if they aren't used anymore they send an unsubscribe message to log off. Additionally, the IAs receive a complete overview of the environment after subscription and get notifications of all changes in the environment. This information is used to create and update the 3D user interface. [37] describes this process in detail.

### *4.4.1 Message format of the AmIS protocol*

The messages of the AmIS communication protocol have a uniform structure which is shown below. XML has been chosen as messaging language

```
<identifier>
        <mid>...</mid>
        <uid>...</uid>
        <key>...</key>
        <value>...</value>
        <situation>...</situation>
</identifier>
```

The whole payload is framed by an identifier tag that represents the type of the message. In this tag the elements *mid*, *uid*, *key*, *value* and *situation* provide all necessary information that is transported. The *mid* field holds the unique message ID which is generated when the message is created. Each message can be referenced

**Table 1:** *List of abstract action types used in AmIS*

| | |
|---|---|
| Action_ON | Action_GENERIC |
| Action_OFF | Action_GOTOCH |
| Action_START | Action_RESIZE |
| Action_NEXT | Action_FREEZE |
| Action_STOP | Action_STANDBY |
| Action_PREV | Action_MUTEPIC |
| Action_DARKEN | Action_MUTESND |
| Action_BRIGHTEN | Action_UP |
| Action_DIMM | Action_DOWN |

by this ID field. The next entry *uid* holds a unique identifier of an AmIS component, a device or a media file. This field is used to identify an entity that is the target of a command or a subscription message. Afterwards, the *key* entry provides the type of notification, the corresponding abstract action or the type of transaction, depending on the used message type. As last element the *situation* field is added. This entry is modeled as a universal field for extra information like a timestamp or anything of interest that might be added in the future.

### *4.4.2 The Action Ontology*

In order to hide the heterogeneity of the various device networking protocols from higher level agents of our system, we provide a device abstraction layer. It makes use of an action ontology to introduce a set of abstract actions. The entire communication in the system uses this action ontology and all the commands targeted to any device go through the device abstraction layer. This unifies the inter agent communication regardless of the fact that the actual physical devices are heterogeneous and most of the time not interoperable.

Table 1 lists the action types supported by the device abstraction layers action ontology. These abstract types are used to form command protocols, which are translated to the actual commands just before sent out to the physical devices. The translation is locally done by the corresponding IoT gateway agents.

For instance Action DIMM is used for an action that controls a dimmer of a device with a specified value range. Other examples are the simple Action ON, Action OFF and Action STANDBY commands that activate or deactivate a device accordingly, with no need for a value range. This level of abstraction is needed to provide abstract and universal access to standard services of devices.

These action types enough information for identifying the properties of an action. It could happen that one action type has different effects on different device types. Therefore, the name of each action of a device is built by a combination of its abstract action type and the device class. This leads to a naming convention that creates, for example, an action name like “Dimmer-Dimm” for the action type “Action_Dimm” on a device with device class “Dimmer”. So it is assumed that each device has a known device type and each action of the device can be mapped to a predefined action type. The translation of these abstract action names to specific device commands is done in the IoT Gateways. This enables each component of AmIS to use the known abstract action names for each device. A list of all abstract actions is given in table 1. Each action has a so called capability connected to it that represents the value range of the action. Currently there are five different capabilities which cover each possible value range [33]:

- **Binary capability:** possible value of 0 and 1
- **OneDimensional capability:** one integer/float as value
- **TwoDimensional capability:** two integers/floats as value
- **ThreeDimensional capability:** three integers/floats as value
- **State capability:** strings like “standby” as value

Another attribute of an action of a device is an effect on a connected device property. So the capabilities are connected to the values of the device property the action affects. So if an action affects a property with strings as value then the capability of the action also uses strings instead of numeric values. The *RoomPropertyEffects* describes which room properties are affected by an action. An example for a projector is shown below:

```
<param name="capabilityentries">
        <Projector_MUTEPIC> <picture/> </Projector_MUTEPIC>
        <Projector_MUTESND> <sound/> </Projector_MUTESND>
        <Projector_SLCT_SRC> <SourceValue/> </Projector_SLCT_SRC>
        <Projector_NEXT> <PageNumberValue/> </Projector_NEXT>
        <Projector_ON> <devicestatus/> </Projector_ON>
        <Projector_OFF> <devicestatus/> </Projector_OFF>
</param>
<param name="propertyentries">
        <PageNumberValue> 15 </PageNumberValue>
        <SourceValue> 40 </SourceValue>
        <sound> 10 </sound>
        <picture> 0 </picture>
        <devicestatus> 0 </devicestatus>
        <DeviceType> Projector </DeviceType>
</param>
<param name="RoomPropertyEffects">
        <Testdevice_DOWN> <lightlevel/> </Testdevice_DOWN>
</param>
```

### 4.5 Dynamic User Interface Creation

At startup, the Environment Manager retrieves a complete environment overview. Therefore, a message is sent to the Environment Monitor which triggers an answer. The process of the user interface preparation is shown in the Figure 7. First, the Environment Monitor retrieves information about the environment in order to

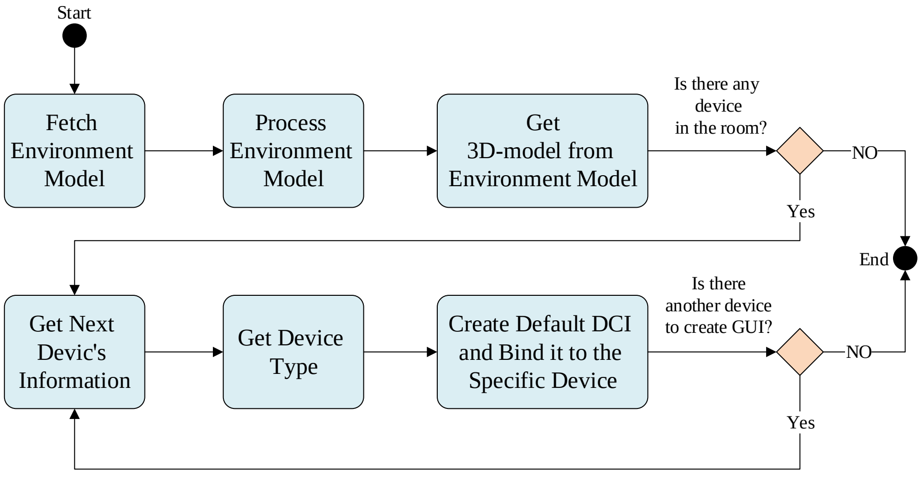


**Fig. 7:** *Model-based User Interface Preparation [44]*

acquire the required data of the physical room, a list of discovered devices, the 3D representation of each device, as well as the device profile, which is needed to retrieve the corresponding Device Control Interface (DCI). These DCI are usual controls, e.g. modeless dialogs, context menus or radio buttons. Each device represented within the 3D-view has a unique ID. This ID allows the user interface to link the 3D object with a physical device which is usually controlled by our system. Since we are dealing with devices with a standardized and well-known states, behavior, and service interfaces, it would not be convenient to use dynamic GUI generation. Simple actions (e.g., turning on or off) are not done through a DCI. Instead, they are done directly through tap on the 3D object itself. More complex devices, e.g., a Smart TV has its own DCI. When the TV's 3D representation is selected, the user is prompted a context menu containing a list of functionalities the complex device provides. For position and orientation information, we use a combination of two indoor positioning systems: the Ubisense system for positioning persons, and a passive RFID based system [40] to gather precise orientation and position information for mobile multimedia devices. Stationary devices such as fixed back projection systems provide their location information by themselves. For each (UPnP) discovered device we look up its position and orientation and disclose devices located in other rooms of the same sub-network. We extend each UPnP device by a UPnP service providing dynamic location information for that device [38]. At the end of the process, the Environment Monitor writes all the information needed to create the user interface into an XML file which is then sent through the Environment Manager to the Interaction Appliance. An XML-file describing an example room is shown below:

```
<room>
<param name="guid" value="ROOM"/>
```

```
<param name="startingx" value="0"/>
<param name="startingy" value="0"/>
<param name="startingz" value="0"/>
<param name="rotateAroundXAxis" value="0"/>
<param name="rotateAroundYAxis" value="0"/>
<param name="rotateAroundZAxis" value="0"/>
<param name="ModelURL" value="11.11.120.13"/>
<param name="propertyentries">
<lighting>25</lighting>
<humidity>12</humidity>
</param>
<room>
```

For each device there is one *service* tag defined that describes the device model. Furthermore, *capabilityentries* contain the supported actions for this device, whereas the tag *propertyentries* describe the specific state for the device. The URL-base is a URL to an archive file that contains the 3D Model and textures as well as the Device Control Interfaces for the described device. The *capabilityentries* describes those attributes of a device that are changed when invoking a specific action. For example the action Dimmer DIMM affects the attribute *lightlevel* of the current device.

```
 <service>
<param name="guid" value="NEC-VT595-123EFA1"/>
<param name="devicetype" value="Dimmer"/>
<param name="startingx" value="40"/>
<param name="startingy" value="17"/>
<param name="startingz" value="66"/>
<param name="rotateAroundXAxis" value="3"/>
<param name="rotateAroundYAxis" value="84"/>
<param name="rotateAroundZAxis" value="25"/>
<param name="URLBase" value="11.11.120.13/deviceModel"/>
<param name="capabilityentries">
 <Dimmer_ON>
  <devicestatus/>
 </Dimmer_ON>
 <Dimmer_DIMM>
  <lightlevel/>
 </Dimmer_DIMM>
</param>
<param name="propertyentries">
 <lightlevel>88</lightlevel>
<devicestatus>1</devicestatus>
</param>
</service>
</body>\
```

### 4.6 Interaction Synchronization

EnvMan must avoid conflicting access to the environment. Within the scope of Human-Environment-Interaction, conflicting access could be the simultaneous access of the implicit interaction system (e.g., a rule engine) and the explicit access to shared resources such as a device or a document. Behind this, conflicting access could also mean executing operations with opposite semantics. For example turning on a heater together with a cooling device is considered as conflicting environment access. Another type of conflicting access is the automatic execution of operations which do not match the user's current intention. Such kind of incorrect automatism could confuse the user and also decrease the overall system usability. In other words, only the right operation executed at the right time could match the user's intention. The AmIS system provides mechanisms to coordinate the coexistence of implicit interaction and explicit user commands. However, describing synchronization details are beyond the scope of this paper, which focuses more on the 3D UI aspect and user evaluations. Readers can refer to [39] for details.

## 5 Proof-of-Concept Implementation

## 6 Evaluation

The evaluation of AmIS has been done by a subjective user evaluation of the software usability based on ISO 9241/110 standard. The user evaluation test has been performed in a professional work office. The persons participating in the evaluation were both trustworthy and competent to perform the evaluation, so that the evaluation results in maximum credibility and acceptance. In the usability test, we analyzed the most important activities of a presentation scenario. We selected test situations and activities involved from our existing situation-concept and a hierarchical task model for presentation scenarios. A total of 46 subjects performed 41 different test cases (e.g., "set up the room for a presentation!" or "present your thesis-slides on the back-projection system!"). Some activities have also been performed using manual controls (e.g. light switch). Before performing an activity, we prepared the room and re-started the interaction systems. Each subject was sufficiently introduced to both types of control systems (explicit and implicit). We avoided to having several test users at the same time in the room. During the inquiry, it became obvious that a verbal explanation at the beginning was desirable and could have positive consequences on further motivation. The subjective usability analysis scrutinized the Perceived Ease of Use (PEU) and the Perceived Usefulness (PU) of the AmIS system from the point of view of end users. In order to analyze the PEU and PU, end users were requested to answer related sets of questions. For data collection, specifically designed questionnaires were given to end users of the AmIS system. We used the ISO-Norm 9241/110 questionnaire, which was derived from the software ergonomic standard DIN EN ISO 9241, Part 10 (German Industry Standard). ISO-Norm 9241/10 Usability is a group of norms that allows evaluating the capacity of an interactive system to offer the possibility to accomplish tasks in an

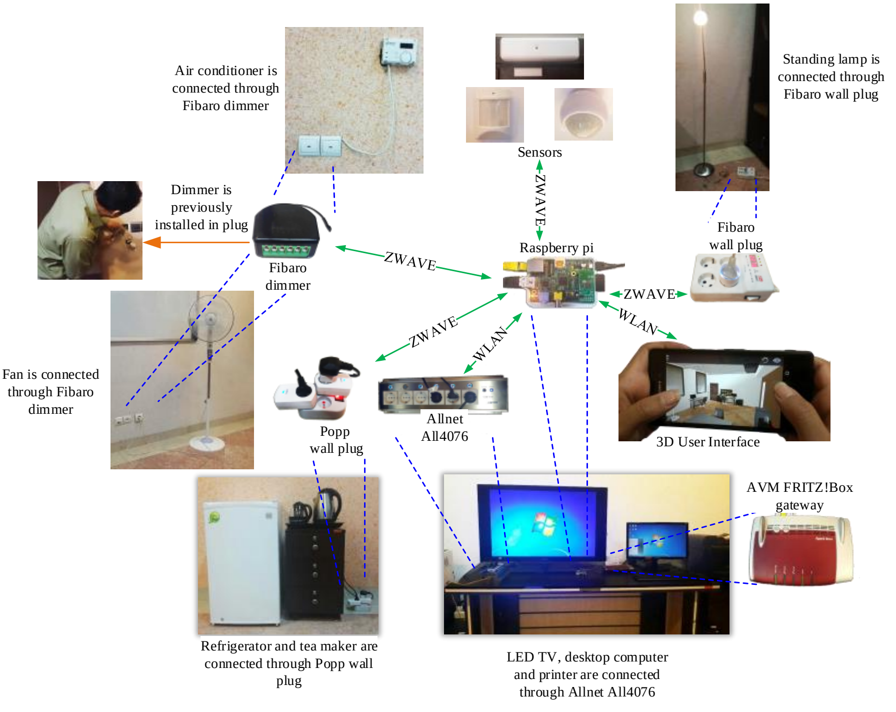


**Fig. 8:** *Smart home implementation using devices, sensors, actuators, controllers and network protocols*

effective and pleasant way. This in effect tests the following additional features: suitability for the task, self-descriptiveness, controllability, conformity with user expectations, error tolerance, suitability for individualization and suitability for learning.

Two questions regarding self-descriptiveness of the application had to be removed since they were related to desktop applications only. The questionnaire is designed for a verbal interview with 30 to 60 minutes per test person. After an introduction, questionnaires were carefully completed. In average, the test subjects' experience with mobile devices and 3D programs was based on a daily routine and at least five hours a week. The first test person required explanation on several points during evaluation which made apparent the need for further question differentiation to eliminate possible ambiguities.

## 7 Results and Analysis

Figure 9 depicts the results of the ISO 9241/110 questionnaire showing the user interface "quality" of AmIS. The figure shows the results for two different user

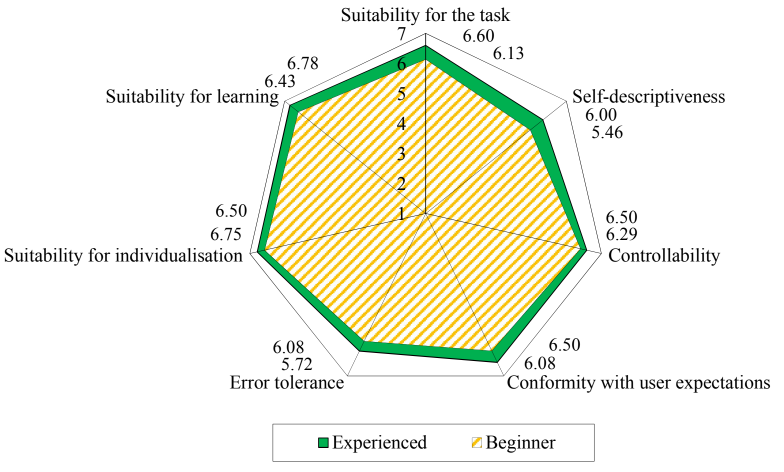


**Fig. 9:** *Result of ISO 9241/10 questionnaire for beginners and experiences users*

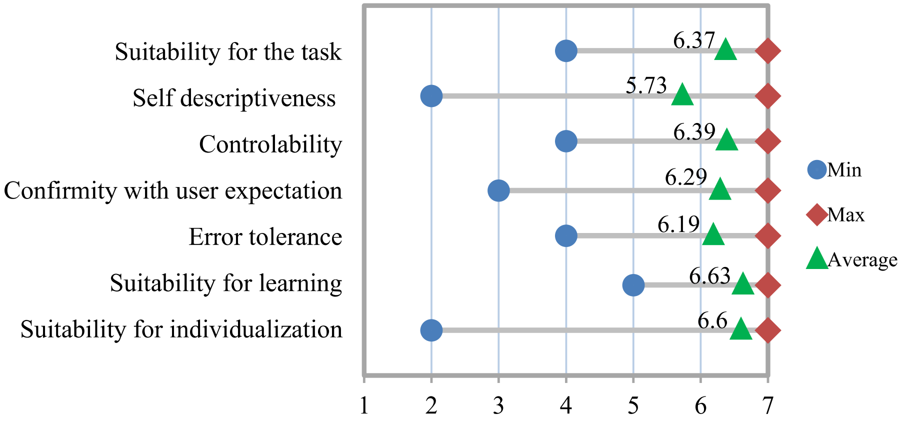


**Fig. 10:** *Mean values, lowest and highest score in seven categories of the ISONorm usability evaluation questionnaire for the AmIS interaction system*

groups: beginners and experienced users. In addition, Figure 10 summarizes the results of the ISO-Norm usability questionnaire in seven categories; “suitability for the task”, “self-descriptiveness”, “controllability”, “conformity with user expectations”, “error tolerance”, “suitability for individualization” and “suitability for learning”. Also an independent sample t-test was conducted to compare experienced users and beginners (cf. table 2). The tests indicate no significant

**Table 2:** *Mean, standard deviation and t-test results for experienced and inexperienced users*

| ISONorm category | Experienced user | Inexperienced user | t-test result |
|---|---|---|---|
| Suitability for the task | $M = 6.61$, $SD = 0.21$ | $M = 6.16$, $SD = 0.34$ | $t(13) = 1.69$, $p = 0.113$ |
| Suitability for learning | $M = 6.68$, $SD = 0.10$ | $M = 6.29$, $SD = 0.24$ | $t(12) = 1.89$, $p = 0.083$ |
| self-descriptiveness | $M = 6.16$, $SD = 0.43$ | $M = 5.59$, $SD = 0.35$ | $t(14) = 1.82$, $p = 0.090$ |
| Controllability | $M = 6.58$, $SD = 0.1$ | $M = 6.26$, $SD = 0.43$ | $t(11) = 1.17$, $p = 0.26$ |
| Conformity with user expectation | $M = 6.52$, $SD = 0.32$ | $M = 6.06$, $SD = 0.32$ | $t(14) = 1.60$, $p = 0.131$ |
| Error tolerance | $M = 6.08$, $SD = 0.34$ | $M = 5.70$, $SD = 0.45$ | $t(14) = 1.18$, $p = 0.255$ |
| suitability for individualization | $M = 5.75$, $SD = 0.21$ | $M = 6.5$, $SD = 0.57$ | $t(12) = 0.79$, $p = 0.440$ |

difference between beginners and expert users. In other words, the presented interaction system is easy to use and suitable for beginners and experts at the same time.

A closer look at the data reveals that the categories “suitability for learning” and “suitability for task” receive a high grading which is very important in this context (cf. Fig. 10). The high grade on these categories address the intuitiveness requirement of the AmIS interaction system. Controllability reached a mean value of 6.39 points and suitability for the task, 6.37 points. By comparison, suitability for self-descriptiveness rated worst at 5.73 which is still a high score according to the ISO-Norm scores. We could say that the study achieved a good overall rating of the AmIS software system and its 3D user interface. To summarize all ratings above 6.00 the majority of users think: AmIS is “easy to use”, “offers all functions to solve tasks efficiently”, “uses comprehensible metaphors”, “offers good opportunities to stop the task and continue at the same point later on”, “allows to change easily between UI parts and menus”, “can be used in a consistent way”, “requires little time to learn”, “encourages to try new functions”, “does not require the user to remember many details” and “is designed in a way that things you learned once are memorized well”. All of this confirms our main projections for the development of AmIS. The category suitability for the task got grade 6.37. It got this grade since some users had problem on some tasks such as shutter movement using the AmIS. For example, users need to navigate the 3D view and make the view bigger in order to perform the shutter movement tasks. The group of users who were beginner to touch screen based mobile 3D navigation usually had some problems to perform this task. However, this task was easy to do for the experienced users (cf. Fig. 9).

The above-mentioned problem can be divided to two sub-problems: device selection and device control. The device selection problem of novice users can be improved through the following solutions:

Scroll bar: by adding a 2D scroll bar on both sides of the user interface, the novice users could easily navigate the view and find their desired device to control.

“Adaptive Navigation”: depending on the selected media type, the 3D scene camera moves to a suitable device which supports the rendering of the specific media type. For examples once the user selects a powerpoint slide and holds on it, then the scene moves to a TV. If the user still presses the stylus on the selected powerpoint slide then the interfaces moves to the next suitable device and so on. It should also be possible to look up devices based on a selected activity (Preset Actions) such as “Lighting”, “HVAC”, “Printing”, etc. Preset Actions are listed on the bottom left part of the AmIS user interface.

The device control problem of novice users can be improved by applying the following solutions:

- 2D device control interfaces outside the 3D view: through using a two dimensional device control interface. When the user taps on a device, its control interface will be activated.
- Goal-based interaction: by using preset actions the user can just express his goals, e.g. “darken the room”. The AmIS system decides how to achieve the goal. The required plan/strategy is contained within the preset action. Preset Actions are stored on the cloud. For the above example “darken the room”, the strategy could be to move down all the shutters in the room, to darken all the dimmers and switch off the lights. Location-aware device discovery is required in order to select the required set of devices and to perform the right actions on them to achieve the requested effect. For more details on location aware device discovery users might refer to [38].

## 8 Conclusion

In this paper, we proposed a mobile 3D UI application for interaction with ambient intelligent environments. We justified our Mixed-Initiative-Interaction technique, and demonstrated through a prototype evaluation its effectiveness. Some shortcomings were also identified which will be addressed in future work.

## References

1. Allnet homepage. http://www.allnet.de/en/
2. Ant+ homepage. http://www.thisisant.com/
3. Digital living network alliance. http://www.dlna.org
4. FRITZ!Box, AVM International. http://en.avm.de/products/fritzbox/
5. Knx association. www.knx.org/
6. Plugwise homepage. https://www.plugwise.com/
7. Raspberry pi homepage. https://www.raspberrypi.org/

8. Upnp forum home page and standards specs. http://upnp.org/
9. Zwave homepage. http://www.zwave.com
10. "IEEE Draft Standard P802.11" (1996) Wireless LAN
11. "IEEE 802.3ah" (2004) Ethernet in the First Mile
12. Aart EH, Encarnação JL (2006) True Visions: The Emergence of Ambient Intelligence. Springer-Verlag
13. Aarts E, Grotenhuis F (2011) Ambient intelligence 2.0: Towards synergetic prosperity. Journal of Ambient Intelligence and Smart Environments 3(1):3–11
14. Aarts E, Wichert R (2009) Ambient intelligence. Springer
15. Allouch SB, van Dijk JA, Peters O (2009) The acceptance of domestic ambient intelligence appliances by prospective users. In: Pervasive Computing, Springer, pp 77–94
16. Balta-Ozkan N, Davidson R, Bicket M, Whitmarsh L (2013) Social barriers to the adoption of smart homes. Energy Policy 63:363–374
17. Bilgin A, Hagras H, Malibari A, Alghazzawi D, Alhaddad MJ (2013) A computing with words framework for ambient intelligence. In: Systems, Man, and Cybernetics (SMC), 2013 IEEE International Conference on, IEEE, pp 2887–2892
18. Bonino D, Corno F (2011) What would you ask to your home if it were intelligent? exploring user expectations about next-generation homes. Journal of Ambient Intelligence and Smart Environments 3(Number):111–126
19. Burmeister D, Altakrouri B, Schrader A (2015) Ambient reflection: Towards self-explaining devices. Large-scale and Model-based Interactive Systems p 16
20. Chen YH, Zhang B, Tuna C, Li Y, Lee EA, Hartmann B (2013) A context menu for the real world: Controlling physical appliances through head-worn infrared targeting. Tech. rep., DTIC Document
21. Gilman E, Davidyuk O, Su X, Riekki J (2013) Towards interactive smart spaces. Journal of Ambient Intelligence and Smart Environments 5(1):5–22
22. Hervás R, Bravo J, Fontecha J, Villarreal V (2013) Achieving adaptive augmented reality through ontological context-awareness applied to aal scenarios. Journal of Universal Computer Science 19(9):1334–1349
23. Hoff KA, Bashir M (2014) Trust in automation integrating empirical evidence on factors that influence trust. Human Factors: The Journal of the Human Factors and Ergonomics Society p 0018720814547570
24. Hossain MA, Shirehjini AAN, Alghamdi AS, El Saddik A (2013) Adaptive interaction support in ambient-aware environments based on quality of context information. Multimedia Tools and Applications 67(2):409–432
25. Kashyap H, Singh V, Chauhan V, Siddhi P (2015) A methodology to overcome challenges and risks associated with ambient intelligent systems. In: Computer Engineering and Applications (ICACEA), 2015 International Conference on Advances in, IEEE, pp 245–248
26. Loeffler D, Hess A, Maier A, Hurtienne J, Schmitt H (2013) Developing intuitive user interfaces by integrating users' mental models into requirements engineering. In: Proceedings of the 27th International BCS Human Computer Interaction Conference, British Computer Society, p 15

27. Mattern F (2003) Vom verschwinden des computersdie vision des ubiquitous computing. In: Total Vernetzt, Springer, pp 1–41
28. Mostafazadeh A, Shirehjini AAN, Daraei S (2015) A meta user interface for understandable and predictable interaction in aal. In: Human Aspects of IT for the Aged Population. Design for Everyday Life, Springer, pp 456–464
29. Nazari Shirehjini A (2007) A multidimensional classification model for the interaction in reactive media rooms. Human-Computer Interaction HCI Intelligent Multimodal Interaction Environments pp 431–439
30. Oduor E, Neustaedter C (2014) The family room: A multi-camera, multi-display family media space. In: Proceedings of the Companion Publication of the 17th ACM Conference on Computer Supported Cooperative Work & Social Computing, ACM, pp 289–292
31. Portet F, Vacher M, Golanski C, Roux C, Meillon B (2013) Design and evaluation of a smart home voice interface for the elderly: acceptability and objection aspects. Personal and Ubiquitous Computing 17(1):127–144
32. Poslad S (2011) Ubiquitous computing: smart devices, environments and interactions. John Wiley & Sons
33. Reetz AA Adrian; Nazari Shirehjini (2008) Realization of a upnp agent for controlling adaptive multimedia environments. Master's thesis, Technische Universität Darmstadt
34. RS232-C ES, Interface between data terminal equipment and data communications equipment employing serial binary data interchange
35. Sadri F (2011) Ambient intelligence: A survey. ACM Computing Surveys (CSUR) 43(4):36
36. Schmitt F, Cassens J, Kindsmüller MC, Herczeg M (2011) Mental models of ambient systems: a modular research framework. In: Modeling and Using Context, Springer, pp 278–291
37. Shirehjini AAN (2004) A novel interaction metaphor for personal environment control: Direct manipulation of physical environment based on 3d visualization. Computers & Graphics 28(5):667–675
38. Shirehjini AAN (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, pp 207–212
39. Shirehjini AAN (2008) Interaktion in Ambient Intelligence: Konzeption eines intuitiven assistenten zur ganzheitlichen und konfliktfreien Interaktion in adaptiven Umgebungen. PhD thesis, Technische Universität Darmstadt
40. Shirehjini AAN, Yassine A, Shirmohammadi S (2012) An rfid-based position and orientation measurement system for mobile objects in intelligent environments. Instrumentation and Measurement, IEEE Transactions on 61(6):1664–1675
41. Shneiderman B (1997) Direct manipulation for comprehensible, predictable and controllable user interfaces. In: Proceedings of the 2nd International Conference on Intelligent User Interfaces, ACM, pp 33–39

42. Vacher M, Lecouteux B, Portet F (2015) On distant speech recognition for home automation. In: Smart Health, Springer, pp 161–188
43. Zeidler C, Lutteroth C, Weber G (2013) An evaluation of stacking and tiling features within the traditional desktop metaphor. In: Human-Computer Interaction–INTERACT 2013, Springer, pp 702–719
44. Zoller AA Martin; Nazari Shirehjini (2008) Realization of a upnp agent for controlling adaptive multimedia environments. Master's thesis, Technische Universität Darmstadt