

Synchronized Realities

Christoph Stahl*, Jochen Frey, Jan Alexandersson and Boris Brandherm

DFKI GmbH, Saarbrücken, Germany

E-mail: first.last@dfki.de

Abstract. Ambient Intelligence (AmI) promises future environments that support the well-being of their inhabitants through comfort and assistive environments. For the elderly users, Ambient Assisted Living (AAL) scenarios are targeting safe and independent living at home. Especially the integration of home automation components, such as intelligent light, window, and climate controls seem promising for such scenarios. In this paper, we present a new development method that is based on a detailed three-dimensional model and show how the model can be used for designing and evaluating assistive environments. Such a system has been developed and deployed. We present an architectural foundation of how Universal Remote Console technology can be applied to synchronize the virtual environment model with the real world. Furthermore, we extend previous work on Dual Reality by introducing the concept of Synchronized Realities that includes the synchronization between remote AmI environments. We will discuss application scenarios and show two demonstrators that we have implemented based on the proposed development method.

Keywords: Intelligent environments, ambient assisted living

1. Introduction

In the scope of the DFKI Competence Center Ambient Assisted Living¹ (CCAAL) we aim to enhance the quality of life of older people through technology. According to the European Ambient Assisted Living (AAL)² research initiative, this includes increasing their autonomy, enhancing their security, and preventing isolation by staying socially connected. Our long-term vision is to promote an accessible intelligent environment based on open standards and architectures and innovative solutions where everyone can continue to play a role in society.

In this paper, we look at the well-being of senior citizens from two perspectives, namely i) physical and cognitive assistance; and ii) social connectedness.

Assistance systems require means to sense activity in the home in order to reason about their users' plans and goals and at least some control over the environment to proactively take supportive actions on behalf of the user. Today, home automation standards already

provide a valuable interface to the environment. Ecological ambitions to save energy foster progress in the field of building automation with intelligent light and climate controls. New wireless sensors and actuators, e.g., offered by the EnOcean alliance³, dramatically reduce the cost and effort of installing such systems. Likewise, home appliances have already become wired to the Net. For example, we use kitchen appliances from Siemens that communicate via the proprietary `serve@home` protocol with a server via power line. For the future, standards like HomeGrid Forum⁴ or DigitalStrom⁵ promise that nearly all electronic devices in our homes will be accessible on the Web with high bandwidth using existing wires. This connectivity allows assistance applications to reliably infer many activities at home based on the use of related devices and artefacts. Vice versa, the assistance system can remind the user to finish a plan, suggest alternative actions, or fully automatically program appliances according to the user's goals.

* Corresponding author. E-mail: christoph.stahl@dfki.de.

¹<http://ccaal.dfki.de>

²<http://www.aal-europe.eu/>

³<http://www.enocean-alliance.org/>

⁴<http://www.homegridforum.org/>

⁵<http://www.digitalstrom.org/>

Social connectedness concerns the experience of belongingness and relatedness with others. Recent research in psychosomatic medicine indicates that social connectedness plays an important role to reduce the risk for a stroke or heart disease. Besides face to face contact, connectedness has been traditionally achieved by telephone and mail. Today, communication can also take place using e-mail, instant messengers, and to a large extent using social networks, e.g., Facebook, to share photos and news. Besides explicit communication, connectedness can also be experienced through awareness systems that inform people about the presence (location) and activities of their friends and relatives. Awareness systems can be based on displays that are designed for ambient, peripheral information presentation, such as the Digital Family Portrait [29]. Whereas displays require users to input information that is to be remotely presented, tangible interfaces can support symmetric interaction. The SnowGlobe [30] is designed as pair of coupled devices that are able to sense the motion of people in their proximity. If one globe senses activity, the remote globe lights up and shows snow fluttering. Tsujita et al. [22] go even one step further and synchronize real-world appliances, such as two TV sets, in order to support remote awareness over long distances like for couples that live separately. The ongoing technical progress towards connectivity and automation of home appliance will open new opportunities to couple interactive devices with each other beyond the boundaries of our homes. We believe that this approach will enable new ways of non-verbal, ambient communication between friends and family that is worthwhile to investigate.

The design and implementation of assistance and awareness systems pose many challenging issues that need to be solved. Regarding the user interface between the human and smart environment, usability is a primary concern, particularly for older users. The cognitive effort that is required to learn how to operate an assistance system must be less than the perceived utility, otherwise the system is likely to be abandoned. From a user's viewpoint it is crucial to stay in control of the assistance system, particularly in case of proactive and automated system behavior where the computer disappears into the background. In [20] Streitz et al. suggest to keep "the human in the loop" thus empowering people to make informed decisions and take actions as mature and responsible people in control.

In the remainder of this paper, we present a new approach towards assistance and awareness systems that exploits the duality of automated homes; the sta-

tus of devices and appliances in the real world on the one hand, and their digital representation on the other hand. If the digital representation is embedded in a virtual world, i.e., a 3-D model that reflects the actual physical environment, we call this a Dual Reality (2R) according to Lifton [8]. Lifton connects real and virtual environments, so that they can mutually reflect and influence each other. Similarly, Streitz and Wichert [21] put hybrid symmetric interaction on their research agenda, meaning that both worlds maintain consistency. In our living labs in Bremen (BAALL) and Saarbrücken (INTELLIGENT KITCHEN), we use home-automation technology to synchronize the status of the real environment with a virtual model thereof, so that the virtual world reflects the real lab situation. Vice versa, actions performed in virtual reality (e.g., to turn on a light by touching a virtual lamp) have a similar effect on the physical environment. We believe that for many users our approach provides a cognitively more adequate way of interacting with their environment. Finally, we investigate how geographically distant environments can be synchronized in order to create the sensation of virtual presence in conjoined environments, i.e., to let users feel as if other family members were present and that they will keep an eye and know if something is wrong.

The paper is structured as follows. In Section 2 we will have a closer look on related work in terms of smart environments, synchronized appliances, and development tools and methods. Section 3 introduces our new concept of Synchronized Realities (SR) that abstracts and unifies existing and distinct approaches of dual reality and synchronized objects. We highlight how the influence between real and virtual worlds can be applied for intuitive monitoring and control, and pervasive social connectedness through synchronized appliances. In Section 4 we outline the stages of our development method GOAL (Geometry-Ontology-Activity Model) that we use to design and develop our AAL labs. GOAL combines activity-based requirements with a detailed 3-D model that is supported by our 3-D modeling toolkit YAMAMOTO. In Section 5 developers learn how the synchronization between realities can be implemented on the basis of existing development tools, methods and standards. Particularly, we introduce how we enhanced the Universal Remote Console (URC) standard (ISO 24752) by a synchronization module. Section 6 discusses a variety of possible application domains for SR that are motivated by our daily experience with AAL-related research projects in cooperation with industrial partners. In Sec-

tion 7 we describe two demonstrators that we have developed in the context of the INTELLIGENT KITCHEN and BAALL apartment as proof of concept. We close the paper with a summary and outlook in Section 8.

2. Related work

Smart environments

One of the first smart environments, the Intelligent Conference Room, has been realized in the MIT project Oxygen⁶ in 1999 with the goal to make computing more human-centered and user-friendly, and to show how computing can help to assist in everyday tasks. The Aware Home⁷ is a three-story living lab at Georgia Tech that aims at awareness of its occupants' whereabouts and activities, and to provide services to its residents that enhance their quality of life or help them to maintain independence as they age. Research in the Philips CareLab [31] is focused on applications for seniors, such as lifestyle assistance, cognitive stimulation, and social connectedness. A range of distributed sensors register activities, which are processed by a context-aware reasoning engine that identifies potential critical incidents, and, in response, alerts care centers or relatives. Similarly, the Fraunhofer Assisted Living Laboratory⁸ has been instrumented with a large variety of ambient sensors (lights, switches, doors, sockets) in order to recognize unusual situations from context by a rule-based activity recognition framework.

Synchronized homes for remote awareness

In our modern society it is common that couples and families have to live separately due to their jobs or studies. Several research projects have dealt with the issue to enhance traditional communication through means of implicit, ambient information channels that provide a sense of awareness of what others are doing in remote places.

Tsujita et al. [22] explore the use of common, day-to-day items to communicate everyday actions. They note that although inexpensive communication by phone and instant messaging is available today, many romantically involved couples, separated by long distances, don't feel that they adequately keep in touch. Tsujita's SyncDecor system involves the synchronization of pairs of daily appliances such as lights, trash

boxes, and TVs. When a person turns on his/her light, the light of his/her partner also gets turned on immediately. These simple interactions provide subtle awareness of daily activities and ease the need for explicit communication through frequent calls or messages, which can be annoying.

In [23] the authors propose the InPhase system to facilitate remote awareness by communicating the occurrence of coincidental, simultaneous actions of both parties, for example, when they are both opening doors, windows or curtains, or watching the same TV channel at the same time. The system provides an ambient indicator that is intended to motivate communication by phone or messaging. The system allows its users to learn when similar actions happen in a natural day-to-day setting, thus enhancing intimacy and closeness. It also preserves privacy, as only similar actions are revealed.

Tool support for SmartHomes

The home automation industry provides standards, such as KNX⁹, and support tools, such as ETS, that support the programming of devices, e.g., which switch controls which light. Other companies provide plugins to visualize and manipulate the state of the environment for debugging. However, these tools are oriented towards the hardware level and not intended for interior designers. The eHOMESIMULATOR [1] is a research tool for the simulation of smart home environments. Application examples are automated lights or music that follows their listeners from room to room. According to [9] the designer's task consists of: i) specification; ii) configuration; and iii) deployment of these components (SCD-process). A similar tool is TATUS [13] that enables developers of ubiquitous computing systems to overcome the cost and logistical issues of real instrumented environments with a 3-D simulation. In [25] projectors are used to visualize the state of a proactive system on the walls (mixed-reality) to provide the users with an understanding of the system; it helps users to debug the rules in case of undesired behavior.

3. Synchronized Realities

We propose to ground the design, development process, and also the user interface of assistance- and awareness systems on a detailed 3-D model of the environment and its instrumentation with sensors and ac-

⁶<http://aire.csail.mit.edu/>

⁷<http://awarehome.imtc.gatech.edu/>

⁸<http://www.aal.iiese.fraunhofer.de>

⁹<http://www.knx.org>

tuators. The intended result of the development process is that the realized smart environment equals the virtual model from the design phase. As an emergent property, Dual Reality (2R) can be realized, which has been defined by Lifton [8] as follows:

“An environment resulting from the interplay between the real world and the virtual world, as mediated by networks of sensors and actuators. While both worlds are complete unto themselves, they are also enriched by the ability to mutually reflect, influence, and merge into each other by means of sensor/actuator networks deeply embedded in everyday environments.”

Lifton’s original work aims at the automatic creation of content for online 3-D virtual worlds, such as SecondLife, from real-world sensor streams. Lifton developed sensor nodes that measure light, temperature, motion, sound, and electrical current and placed them at various places in his lab. The building structure of his lab has a virtual representation in SecondLife, where the sensor readings are embodied as dynamic geometric figures, such as ponds and fires, at the respective locations of the sensors.

Figure 1 depicts how the mutual influence between real and virtual worlds can be applied to human-environment interaction in an AAL scenario. The images on the left hand side depict our living lab in Bremen, and the images on the right show our lab in Saarbrücken. The images on top show virtual 3-D models, and the images below show actual photographs of the lab.

As the user pushes buttons to move the kitchen block in Bremen up or down in order to fit their individual needs, the virtual model immediately reflects this change. Vice versa, the user can manipulate the kitchen through their avatar in a virtual world, thus adjusting the real kitchen. The mutual influence is made possible through home automation and Web-enabled appliances, as described in the introduction. Both worlds are connected to the URC framework (see Section 5) which is responsible to keep them in sync. Similar interaction can take place in Saarbrücken, for example it is possible to turn the lights on and off using the real switches or by touching virtual lamps that are shown in the 3-D model.

It is important to mention that the virtual world is conceptually neither a pure visualization of the real world, nor does it act like a remote control. In contrast to typical master-slave configurations, our concept of synchronized realities assumes that both the real and

virtual world are complete and on par with each other. This may seem a bit paradoxical given that the virtual world is just a model of the real world. But if we consider a pair of real places, for example the labs in Bremen and Saarbrücken, the assumption of equality has the advantage that we can extend the synchronization relationship to their virtual models as well. Second Life is a good example where multiple users can participate and interact with their avatars in a shared virtual online world that supports social connectedness.

Another prominent example for synchronized virtual worlds is the Massively MultiPlayer Online Role-playing Game (MMPORGs) World of Warcraft. In the context of AAL, it might be reasonable to place virtual models of distant real places next to each other in the virtual world. Hence users might experience closeness between both places with their avatars in virtuality. However, the immersion into virtual worlds requires a certain amount of experience and technology, at least a PC and display. Whereas Virtual Reality (VR) is highly popular among young people, it is clearly not an appropriate solution for senior citizens.

Hence we propose the synchronization between real places as an alternative to shared virtual worlds in order to provide pervasive social connectedness. The concept of Synchronized Reality can be applied to reflect sensed changes in one place by controlling devices and actuators in the other place. For example, this allows to implement pervasive awareness systems, e.g., by turning on a small light in one place to indicate activity in the other place. If one is willing to accept a higher level of intrusion, synchronized appliances, such as Tsujita’s SyncTV, can also be realized based on the same concept. The synchronization paradigm might also be helpful to automatically switch devices between two remote places. For example, if we pick-up the phone in the office, the phone at home could automatically be set for call forwarding. Likewise, if we turn on the heating at home, the heating in the office can be turned off.

We will see in the following sections, how the URC framework can be applied to connect and access devices and home appliances, and how the synchronization module eases the implementation of such scenarios.

4. Designing and developing smart environments

In this section, we outline our design method for intelligent environments called GOAL [16,17], a struc-

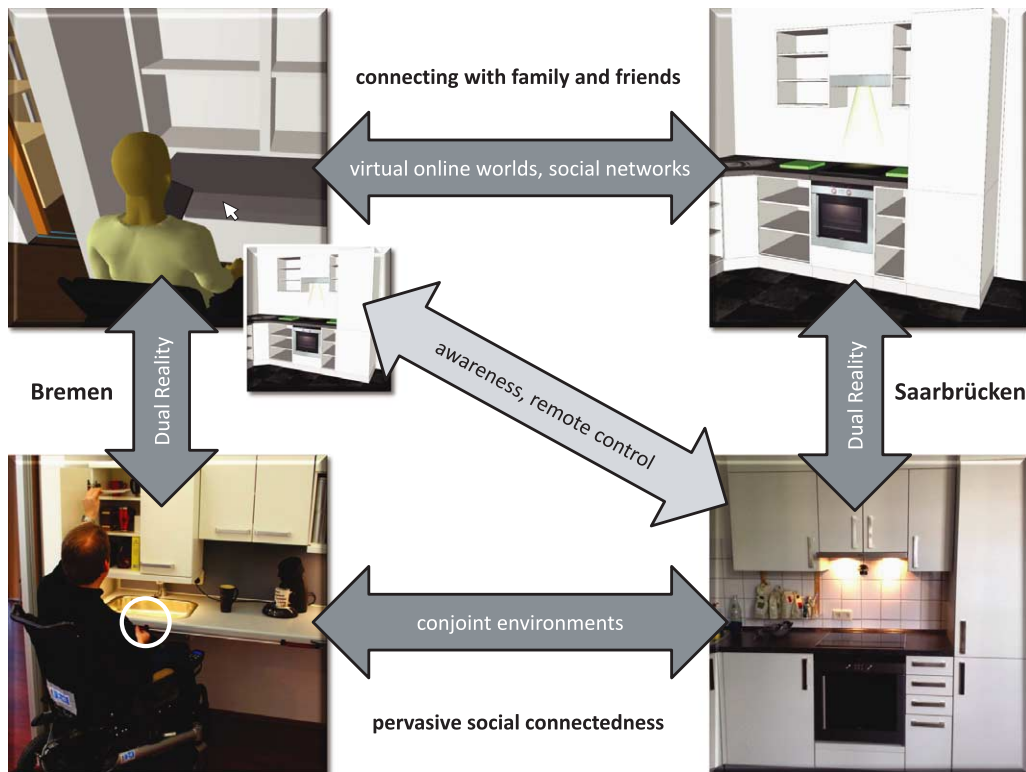


Fig. 1. The concept of Dual Reality (vertical axis) refers to the mutual influence between the real and the virtual world. Both worlds are connected to the Universal Remote Console framework thus keeping them in sync. User actions and device states in one world are reflected in the other world accordingly. The horizontal axis shows the virtual (top) and real (bottom) connections between the living labs in Bremen and Saarbrücken.

tured procedure to analyze the users' needs with respect to the surrounding environment in order to identify useful assistance features. Furthermore, the method supports the designer to make the necessary decisions for the instrumentation of the environment, particularly which sensors and actuators to use and where to place them. GOAL is based on three pillars: i) a geometric environment model; ii) an activity model; and iii) an ontology which provides the symbolic names that tie both models together. We ground our design method on a detailed 3-D model of the environment where the assistance system is about to be installed. All elements of the geometric model have their symbolic counterparts in the ontology. The activity model is derived from Activity Theory (AT) [7] and defines activities that reflect the users' needs. We show how the additional effort to create a 3-D environment model is worthwhile as it can be used after the system's deployment for monitoring and manipulating the state of the real environment in a Synchronized Reality setting.

After an initial assessment of the users' needs with informal methods, the GOAL design method proceeds with the following steps:

(1) Geometric location modeling. YAMAMOTO [18, 19] has been developed for the rapid modeling and visualization of multi-level buildings in 3-D for route finding and indoor navigation applications. Parametric 3-D objects allow to model the building structure with walls, windows, and doors, and the internal furnishing, like shelves, cabinets, and tables. Additionally, artifacts and appliances, like a TV or dishwasher, can be modeled.

(2) Creating an ontological environment model. We create a symbolic representation of the domain, which defines a unique identifier for each subject, artifact, and spatial region of the model. We use the UbiWorld¹⁰ ontology because it provides a Web-based toolkit to view and edit the terminology and instantiations.

¹⁰<http://www.ubisworld.org>

(3) Creating a situated semantic activity model. The users' activities and needs are modeled to organize and structure the informal requirements gathered by elicitation techniques like interviews or ethnographic methods. The subject of the activity/action/operation refers to the actor that performs it. The subject can either be an individual person, or, more generally speaking, a role like *user* or *care giver*. According to AT, artifacts act as a mediator between subject and objects (e.g., a remote control is used to watch TV).

(4) Interaction design. The activity model is analyzed to identify actions and operations that are likely to require assistance. The system designer must figure out how the assistance system could recognize a situation to proactively support the user. The choice for modalities determines how the interaction with the system will look and feel and is crucial for the usability and acceptance of the system. Once sensors and actuators have been chosen, the problem of where to physically place them arises. Particularly optical devices, such as cameras and displays, have a limited field of view that needs to be considered. YAMAMOTO allows the designer to virtually place and evaluate displays from various perspectives and helps to assess how many sensors, e.g., cameras, are required to completely cover an area.

(5) System implementation. Contextual knowledge about activities can be applied to implement activity recognition components. The spatial model can be used for indoor positioning and location-awareness, and the ontology can be applied to share knowledge between applications.

(6) Simulation and evaluation. Instrumenting a real environment with hardware is work and cost intensive, and often implies irreversible damage like drilling holes in walls to mount displays. Sometimes testing on site is not possible at all because of construction work in progress. The simulation of the environment in Virtual Reality aids the designer to verify their concept immediately and efficiently. Important aspects are the visibility of displays and the range of sensors and actuators. Virtual environments can also be used to let the target users experience the system at an early stage in order to verify that the design concept meets their expectations and help to communicate the required hardware effort.

(7) Monitoring and manipulating the environment. Once the system has been deployed it can be synchronized with the virtual model, as introduced in Sec-

tion 1. The model can now be used to visualize the state of the real hardware sensors and actuators. Yet the active implementation of the virtual sensors from the simulator can still be used to manipulate the real environment, for example to open a real door by pushing a virtual button with the avatar.

5. Universal Remote Console

One of the core concepts of our approach is the usage of standards-based open architectures. We propose the industrial Universal Remote Console (URC) standard [5], that focuses on accessible and inclusive user interfaces by allowing any device or service to be accessed and manipulated by any controller [3]. Like other middleware approaches, e.g., AMICO [12], the URC framework provides an adaptive architecture that supports the flexible integration and reuse of heterogeneous software and hardware components, communication protocols and target appliances. By contrast the focus of URC is the notion of *Pluggable User Interfaces* that allows for interfacing arbitrary networked appliances or services with secure, personalized and, perhaps most important, accessible user interfaces [3]. Users can select a user interface that fits their needs and preferences, using input and output modalities, and interaction mechanisms that they are familiar with and work well with them [27]. The first project in Europe using URC technology was *i2home*¹¹, which had the ambitious effort to inject an ecosystem around the industrial URC standard [5] and to introduce URC technology in the field of AAL. Since *i2home* started, in Europe alone projects with in total 120 partners and an accumulated budget of about 80 million Euro are currently using the URC technology, e.g., VITAL¹², MonAMI¹³, Brainable¹⁴, Smart-Senior¹⁵ or SensHome. Therefore, we are member of the OpenURC Alliance¹⁶, which is an international initiative for exchanging experiences, ideas and, perhaps most importantly, continue the development of the URC standard.

The Universal Control Hub

Figure 2 shows the modular architecture of a Universal Control Hub (UCH) [26]. A UCH is a gateway-

¹¹www.i2home.org

¹²www.ist-vital.org

¹³www.monami.info

¹⁴www.brainable.org

¹⁵www.smart-senior.de

¹⁶www.openurc.org

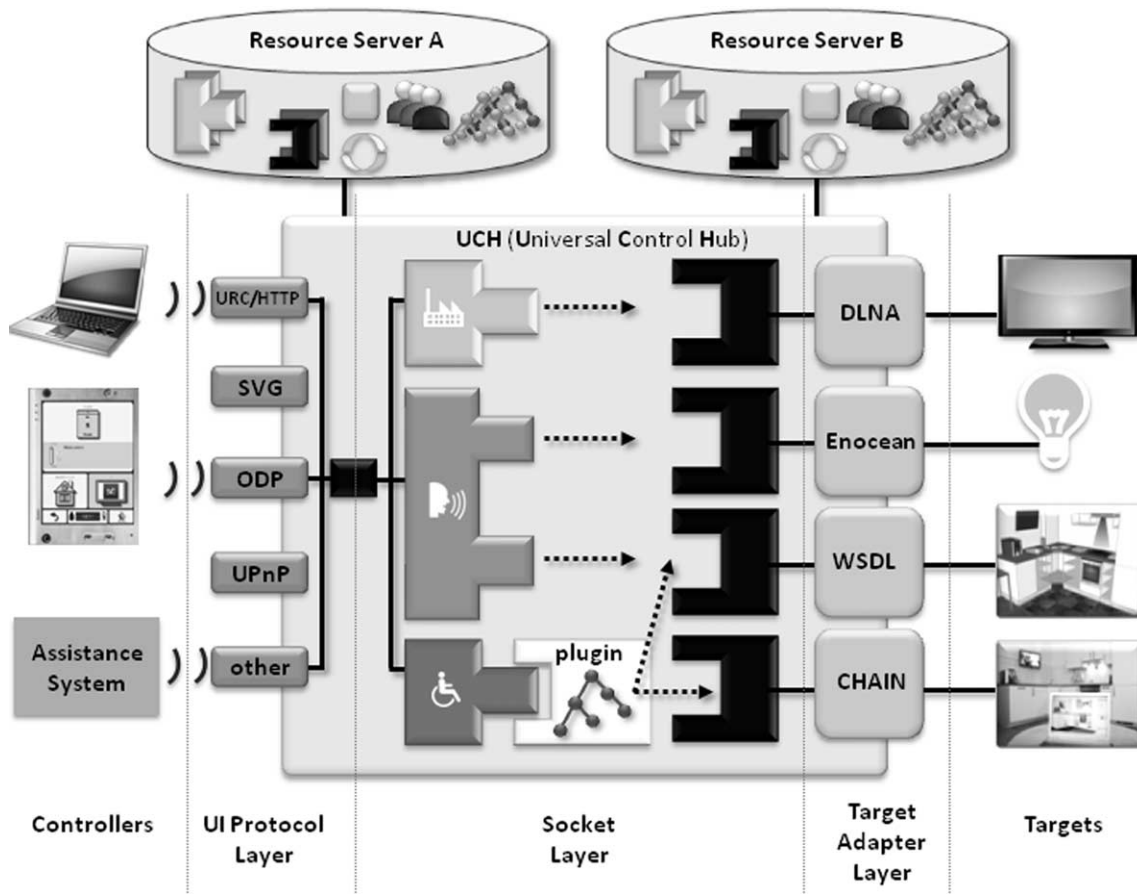


Fig. 2. The URC framework provides an adaptive architecture that supports the flexible integration and reuse of heterogeneous software and hardware components, communication protocols and target appliances and therefore allows any target device to be accessed and manipulated by any controller device.

based architecture implementing the URC standard thus managing the communication between controller and target appliances: i) a *Controller* is any device for rendering the abstract user interface, e.g., TV, touch screen or smartphone; ii) a *Target* is any networked device or service intended to be controlled or monitored, e.g., kitchen appliance, home entertainment or eHealth devices; and iii) a *Resource Server* is a global service for sharing *Pluggable User Interfaces* and various resources necessary for interacting with the targets, e.g., language resources, icons or presentation templates defining the arrangement of interface elements. Currently, a pilot resource server is being operated by dot UI¹⁷. The benefit of this approach is that it is possible to deploy consistent and, particularly, accessible interfaces which are tailored to particular users and their

specific needs. The UCH architecture mainly consists of three layers: i) the User Interface Protocol Layer is responsible for defining User Interface Protocol Modules (UIPM) which specify the communication protocols for the controllers, e.g., HTTP, SOAP, SVG, UPnP or proprietary solutions; ii) the User Interface Socket Layer defines the standardized *User Interface Socket* (UIS) that serves as a common contract between controllers and targets (see Section 5); and iii) the Target Adapter Layer manages the grounding and communication with actual targets, e.g., a DLNA television, Enocean lighting or CHAIN based kitchen appliances.

Standardized contract between user interface and target

The standardized definition of the *User Interface Socket* (UIS) describes the input and output behaviour of a specific target device on an abstract level and therefore is the common model of all user interfaces

¹⁷www.dotui.com

```

<variable id="temperature" type="xsd:double">
  <dependency>
    <write>>false()</write>
  </dependency>
</variable>

<command id="cancelReset" type="
  uis:basicCommand">
  <dependency>
    <relevant>uis:hasDefinedValue('
      checkReset') and uis:value('
      checkReset') eq 'active'</relevant>
    <write>uis:hasDefinedValue('checkReset'
      ) and uis:value('checkReset') eq '
      active'</write>
  </dependency>
</command>

<notify id="CheckReset" category="alert">
  <dependency>
    <explicitAck> false() </explicitAck>
    <acknowledge>
      (uis:hasDefinedValue('confirmReset')
        and uis:value('confirmReset')
        eq 'done')
      or (uis:hasDefinedValue('cancelReset'
        ') and uis:value('cancelReset')
        eq 'done')
    </acknowledge>
  </dependency>
</notify>

```

Fig. 3. The *User Interface Socket* (UIS) describes the input and output behaviour of a specific target device on an abstract level and contains a flat set of socket elements that provide a synchronized communication channel to the controlled device and its current state. The example shows an example socket for a thermostat target device depicting a variable for the current temperature, a notification for requesting a user acknowledge for resetting the device and a command for cancelling.

and communication protocols. A UIS contains a flat set of socket elements that provide a synchronized communication channel to the controlled device and its current state. Figure 3 shows example socket elements which are either variables, commands or user notifications. The description also specifies how socket elements depend on each other, for example that the commands for confirming or cancelling the request are only relevant when the request notification is active. More advanced dependencies can also be described through the notion of pre- and postconditions. Clearly, the UIS does not provide enough information for creating a nice-looking and accessible user interface. What's missing are concrete instructions how to build the user interface, what widgets to use, how to arrange and structure them and finally the language of labels and messages. These optional resources are pro-

```

<task id="SwitchOffDevices" relevant="true">
  <subtasks id="SwitchOffDevicesSteps" ordered
    ="true">
    <!-- Turn off TV -->
    <step name="turnOffTelevision" task="
      TurnOffTelevision" />
    <binding slot="$turnOffTelevision.
      deviceState" value="'OFF'" />

    <!-- Turn off Air Condition -->
    <step name="turnOffHVAC" task="TurnOffHVAC"
      />
    <binding slot="$turnOffHVAC.deviceState"
      value="'OFF'" />

    <!-- Turn off Hood -->
    <step name="turnOffHood" task="TurnOffHood"
      />
    <binding slot="$turnOffHood.deviceState"
      value="'OFF'" />
  </subtasks>
</task>

```

Fig. 4. An example task-model description based on ANSI/CEA-2018. The task-model contains three steps which are executed sequentially and switch off three different URC target devices.

vided by a *Pluggable User Interfaces* [24], a channel-specific user interface description that plugs into a particular UIS. The socket can then be rendered on some controller thus giving the abstract UIS a concrete implementation, or, in other words, plugging the socket. The concrete user interface connects to one or multiple sockets in two directions: first, getting and representing the values that reflect the current state of the target, and second, requesting changes in the target's state through variable changes and command invocations. On the backend side each target appliance is represented by a dedicated target adapter that is responsible for the grounding of abstract socket elements with any specific network protocol. The UCH architecture offers a flexible way of connecting different user interfaces with any UIS and therefore with any connected target device. Multiple controllers and targets can be attached, exchanged and detached at runtime.

An open plugin infrastructure

There are several UCH implementations available, both open-source and commercial. The UCH described here is based on an OSGI infrastructure for integrating loosely coupled modules in the form of OSGI bundles, e.g., UIPM's, Targets and *Pluggable User Interfaces*. In addition, our UCH implementation includes a dynamic discovery mechanism for optional

plugin extensions (see Fig. 2). With i2home's focus on activity management a task-model plugin based on ANSI/CEA-2018 [14] has been integrated. The plugin and its internal task-model engine allows for the implementation of home automation and task-based scenarios, e.g., leaving home or preparing a meal. The task-model plugin is itself represented by means of a UIS and therefore is accessible by all attached controller appliances. Figure 4 shows an example task-model description for automatically switching of different URC target devices. During the i2home project several home automation and task based scenarios have been implemented, tested and evaluated with real users [6].

The synchronization of different realities is implemented by an additional synchronization plugin operating on the abstract UIS. For appliances in different realities, e.g., a kitchen in the real world and one in the virtual world, the UIS defines a common understanding of the properties and functions that are shared between realities. From a technical perspective the only difference between these two worlds is the actual grounding of the socket elements, i.e., *serve@home* communication in the case of the real extraction hood and API access in the case of the virtual YAMAMOTO model. The synchronization module is responsible for managing the values of the pertaining socket instance and for aligning attached targets. Due to the abstract socket description, URC defines a common bi-directional link between realities and thus perfectly supports the notion of synchronized realities and the proposed GOAL methodology. For the designer of intelligent user interfaces or assistance systems, it makes no difference whether the target environment is virtual, real or a synchronization of several worlds.

6. Synchronized Reality application scenarios

In the following paragraphs, we will discuss possible application scenarios for Synchronized Realities, concerning control, awareness, and the debugging of environments:

Taking control of local and remote places

Proactive systems aim to replace explicit interaction with the user through automated behaviour based on context and adaptation. However, such systems are likely to fail in certain situations that could not be foreseen by the developers. In that case, user interfaces that put the user in control are required, either to manually take actions, or to adapt the system behavior ac-

ordingly. We assume that (virtual or physical) three-dimensional models let the user directly manipulate their environment in a more intuitive and cognitively adequate way than symbolic navigation through lists and menus.

For example, in [11, p. 96] Norman discusses the mapping problem of light switches: which switch controls which light? It is difficult to produce a full natural mapping between a linear row of switches and irregularly placed lights in a room. To improve usability, Norman designed a control box where the switches are placed according to the position of the lights on a floor plan.

Virtual models could also allow users for the remote surveillance of their homes; the users could check for open windows and remotely control the heating in order to warm up their house before arrival when returning from a vacation. In a retail scenario, customers could find products in a 3-D online shop by applying their spatial knowledge of the real store.

In combination with positioning systems, virtual environment models could also help the user to visualize the location of others, objects, or assets. Considering Smart Factory environments, [28] describes how mobile devices can support workers at an industrial production plant in their repair jobs. In a SR environment a virtual model could visualize the position of the workers in real time which allows the supervisor to assign the nearest worker for a pending maintenance job in order to reduce delays that are caused by walking distances. Furthermore, the service staff could remotely inspect and control the machines in a virtual model.

Remote awareness and care taking

We have already introduced SyncDecor [22] in Section 2 that synchronizes everyday appliances to support remote awareness through an ambient communication channel, and InPhase [23] that indicates good moments to initiate a conversation. The same concept might be useful in AAL scenarios to foster communication between seniors that live alone and separately from their relatives. While the primary goal of AAL technology is to support independent living at home, it also aims at reducing the care taker's workload by providing them with detailed, up-to date information about their patients. Today, in case of a serious fall, it can take up to 24 hours until the care taker will find their patient lying on the floor, so the patient has to suffer from unnecessary pain and side-effects like dehydration. Whereas video observation is a quite effec-

tive means to recognize emergency situations, it is still time consuming for the care takers to watch the video streams. Furthermore, video observation is ethically difficult as it massively violates the users' privacy. Instrumented environments employ sensors to provide qualitative information about the users' actions, which could be visualized in a dual-reality model on a more abstract level than video recordings. In this context, also small-scale physical models could be interesting as they require neither a PC nor expertise in how to use VR systems.

Debugging smart environments

It must be anticipated that intelligent environments will suffer from similar reliability issues as desktop software applications, since they rely on complex software components for the realization of intelligent behavior that communicate with hardware components from different vendors. Also the relationship between cause and effect of a failure is very complex. Let us consider a rather simple scenario, where a wireless light switch is used to control a lamp. Now assume a typical failure: the user is unable to turn on the lamp by pushing the button. Due to the complex architecture of Intelligent Environments (IE), the problem could occur on four levels: i) Application: The control program might have exited with an exception that was caused by a software bug at runtime; ii) Network: The network might be misconfigured or disrupted by radio interference; iii) Controller: Either the switch or lamp micro-controller might be damaged or suffer from power failure; iv) Physical: The tungsten filament or the contacts of the switch might be broken.

Hence graphical tools are important to inspect the actual status of sensors and actuators in a cognitively adequate manner: Does the switch really send an 'on' message, and does the lamp really receive an 'on' signal? SRs provides an ideal environment to trace system errors, even remotely. The virtual environment would also allow the developer to playback and repeat previously recorded situations of failure, similar to a stack trace of software applications. The SDDL standard could be used for such sensor logs, see [4].

More advanced tools could implement a physical model that simulates cause and effect; such a model could help to analyze a situation by comparing the states of the simulation with the real world. For example, if the measured room temperature is lower than expected from the simulation given the same state of the central heating and control unit, one can deduce that a radiator valve is defective.

Issues and limitations of synchronization

The synchronization of devices and appliances between remote places can be intrusive and implies that both parties must be willing to cooperatively use them and to consider each other's needs. For example, if the TV set is synchronized, both users have to agree on the same channel and mutually respect each other's interests. This may seem strange for people living alone. However the benefit is that the synchronization can create the sensation of virtual presence of the other person. In case of virtual environments, the synchronization allows others to use the VR world to control real devices. Granting permission to control the environment to others via SR is potentially dangerous, i.e., remotely turning on the stove can cause a fire. Users of a virtual interface might not be aware of real-world implications. Hence it is important to secure the environment against errors and abuse.

The reliability of sensors is limited because they can only partially measure the state of the real environment, and readings are generally prone to noise. This can lead to wrong synchronization and in the worst case to frequently alternating changes. Simple sensors are also unable to recognize the cause for physical problems, e.g., a temperature sensor could be covered with clothes, or a door might be blocked by an obstacle. The user of an instrumented environment must be aware of these limitations in order to judge unexpected system states.

7. Demonstrators

In the following, we will describe the synchronization between real and virtual worlds in case of the Bremen Ambient Assisted Living Lab BAALL and the INTELLIGENT KITCHEN. BAALL is an automated 60 m² apartment suitable for the elderly and people with physical or cognitive impairments. It comprises all necessary conditions for trial living, intended for two persons, as shown in the YAMAMOTO 3-D model in Fig. 5. BAALL aims to compensate physical impairments of the user through adaptable furniture, such as a height-adaptable kitchen, and mobility assistance; an intelligent wheelchair and walker navigate their user within the lab. Accordingly, the lab has been equipped with five automated KNX-based sliding doors that open automatically to let the wheelchair pass through. The doors can also be operated manually by pressing buttons or gently pushing the door itself. Furthermore, all lights in BAALL are automated. YAMAMOTO polls

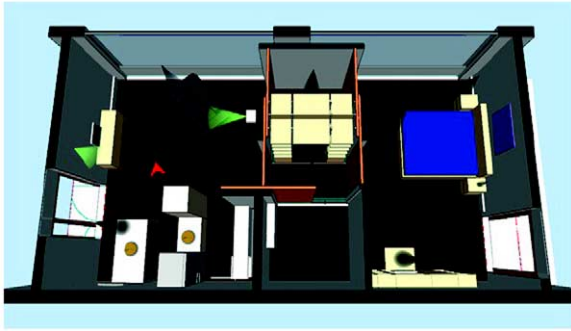


Fig. 5. The lights and sliding doors of DFKI's Ambient Assisted Living Lab (BAALL) in Bremen are synchronized with the 3-D model. They can also be interactively controlled from the model.

the status of the sliding doors and lights in the real environment and visualize it accordingly in 3-D, as shown in Fig. 5. Light sources are visualized as cones that appear either black (off) or bright yellow (on). As the user switches a light, the cone in the model also immediately changes its color. Likewise, the sliding doors appear either open or closed, depending on their true state. In order to control the real world from YAMAMOTO we have extended the geometric scene description by a set of functions for each element. The functions are available through a context menu that is attached to the virtual objects. Hence the real lights can be switched on and off by manipulating the cones in the virtual environment, and the doors can be opened and closed.

The INTELLIGENT KITCHEN is a fully instrumented room that allows for realizing kitchen and living room scenarios. The SEMANTIC COOKBOOK (SC) [15] application utilizes RFID sensors to recognize products and tools on the counter and provides the user with multi-media instructions for cooking recipes. We have fully modeled the kitchen with all its shelves, appliances, and its instrumentation with RFID sensors and a wall-mounted TFT display. Figure 6 shows a photo of the real kitchen that is synced with the YAMAMOTO 3-D model, shown on the laptop from the same perspective. We have focused on networked appliances; the fridge, cooker, oven, hood and dishwasher are connected to the Web via the proprietary *serve@home* protocol. The URC framework synchronizes for example the state of the hood light with the virtual environment model. Furthermore, the TV set (Windows Media Center) has been integrated into the URC framework. As a specialty, we stream the current content from the real TV screen to the virtual display



Fig. 6. The programme on the TV set is reflected in the virtual model (laptop in the foreground). A context menu in the model acts as a remote control to switch channels; also the hood light is in sync between both realities.

in the 3-D world. Hence it is possible to watch the current programme with the avatar in the virtual model. The virtual TV's context menu allows to switch the channels of the TV, which affects both the real and virtual TV set. Referring to our concept of Synchronized Reality (see Fig. 1) this implementation is orthogonal to Tsujita's [22] synchronized real-world TVs. Using URC, it makes no difference whether virtual or real TVs are synced as their implementation is transparent to the system. In a user study described by [22] subjects also suggested to synchronize other devices in the house as well, for example the opening and closing of curtains. Using URC, any connected devices could be synchronized with little effort. For the future, we are planning to extend the visualization with the items on the counter, which are sensed by the SC using the RFID readers.

8. Summary and outlook

We presented the concept of Synchronized Realities that generalizes the idea of Dual Reality to any combination of real and virtual worlds that mutually influence each other, including physical small-scale models. Considering the realization of synchronized realities, we showed how the application of the GOAL development process leads to a virtual model that equals the real environment. Furthermore, we explained how the implementation of the synchronization process is supported by the URC framework. We discussed possible applications for SR in a home scenario, i.e., to manipulate and remote-control automated homes, pro-

vide awareness between people in remote places, and to help debug assistance systems. Further applications comprise smart retail and factory domains. Finally, we presented two actual demonstrators, the BAALL and INTELLIGENT KITCHEN. Both allow the user to visualize and manipulate elements of the real world, such as lights, doors, and a TV set, from the virtual model. For the future, we plan to extend the capabilities of the demonstrators, e.g., include the real-time visualization of movable objects based on RFID sensor data and people that are localized by an indoor-positioning system. By synchronizing our living labs in Bremen and Saarbrücken we want to evaluate what kind of devices and services should be synchronized with respect to user acceptance. Furthermore, we want to evaluate to what extent the smart integration of social connectedness should influence our everyday live, which will become an interesting topic in the field of industrial design. On a larger scale, the synchronization of multiple homes and cars could lead to smart cities.

References

- [1] I. Armac and D. Retkowitz, Simulation of smart environments, in: *Proc. of the IEEE Int. Conf. on Pervasive Services 2007 (ICPS'07)*, IEEE, 2007, pp. 257–266.
- [2] B. Brandherm, S. Ullrich and H. Prendinger, Simulation of sensor-based tracking in Second Life (demo paper), in: *Proc. of 7th Int. Conf. on Autonomous Agents and Multiagent Systems (AAMAS 08)*, ACM, 2008, pp. 1689–1690.
- [3] J. Frey, C. Schulz, R. Nesselrath, V. Stein and J. Alexanderson, Towards pluggable user interfaces for people with cognitive disabilities, in: *Proc. of the 3rd Int. Conf. on Health Informatics (HEALTHINF)*, Springer, 2010, pp. 428–431.
- [4] A. Helal, A. Mendez-Vazquez and S. Hossain, Specification and synthesis of sensory datasets in pervasive spaces, in: *Proc. of IEEE Symposium on Computers and Communications (ISCC 2009)*, IEEE Xplore, 2009, pp. 920–925.
- [5] ISO, ISO/IEC 24752: Information Technology — User Interfaces — Universal remote console — 5 parts, International Organization for Standardization, 2008.
- [6] M. Klíma, U. Diaz, A. Etxaniz, E. Urdaneta, J. Görlich, M. Dubielzig and M. Lindemann, i2home Deliverable D8.4—Final Evaluation Report, 2010.
- [7] A. Leontiev, *Activity, Consciousness, and Personality*, Prentice Hall, Englewood Cliffs, N.J., 1978.
- [8] J. Lifton, Dual Reality: An Emerging Medium, Ph.D. Dissertation, Massachusetts Institute of Technology, 2007.
- [9] U. Norbistrath and C. Mosler, Functionality configuration for eHome systems, in: *CASCON '06*, ACM, 2006, p. 8.
- [10] U. Norbistrath, P. Salumaa and A. Malik, eHome specification, configuration, and deployment, <http://www.ubicomp.org/ubicomp2005/programs/demos.shtml>, 2005.
- [11] D. Norman, *The Design of Everyday Things*, Basic Books, NY, 2002.
- [12] Z. Obrenovic and D. Gasevic, Open Source Software: All you do is put it together, *IEEE Software* **24**(5) (Sept/Oct, 2007), 86–95.
- [13] E. O'Neill, M. Klepal, D. Lewis, T. O'Donnell, D. O'Sullivan and D. Pesch, A testbed for evaluating human interaction with ubiquitous computing environments, in: *Proc. of the 1st Int. Conf. on Testbeds and Research Infrastructures for the Development of Networks and Communities (Tridentcom)*, IEEE Xplore, 2005, pp. 60–69.
- [14] C. Rich, Building task-based user interfaces with ANSI/CEA-2018, *Computer* **42**(8) (2009), 20–27.
- [15] M. Schneider, The semantic cookbook: Sharing cooking experiences in the smart kitchen, in: *Proc. of IE-2007*, The Institution of Engineering and Technology (IET), 2009, pp. 416–423.
- [16] C. Stahl, Spatial Modeling of Activity and User Assistance in Instrumented Environments, Ph.D. thesis, Saarland University, 2009. Online: [urn:nbn:de:bsz:291-scidok-34312](http://scidok.sulb.uni-saarland.de/volltexte/2010/3431/), <http://scidok.sulb.uni-saarland.de/volltexte/2010/3431/>.
- [17] C. Stahl, Modeling and designing user assistance in intelligent environments with the GOAL method, in: *Proc. of Aml 2009 Workshop on Interact. Tech. and Metaphors in Ass. Smart Env. (IntTech)*, 2009.
- [18] C. Stahl and J. Hauptert, Taking location modelling to new levels: A map modelling toolkit for intelligent environments, in: *Proc. of 2nd International Workshop on Location- and Context-Awareness (LoCA)*, LNCS, Vol. 3987, Springer, 2006, pp. 74–85.
- [19] C. Stahl and T. Schwartz, Modeling and simulating assistive environments in 3-D with the YAMAMOTO toolkit, in: *Proc. of 1st Int. Conf. on Indoor Positioning and Indoor Navigation (IPIN)*, Zurich, IEEE Xplore, 2010.
- [20] N.A. Streitz, T. Prante, C. Röcker, D. van Alphen, R. Stenzel, C. Magerkurth, S. Lahlou, V. Nosulenko, F. Jegou, F. Sonder and D. Plewe, Smart artefacts as affordances for awareness in distributed teams, in: *The Disappearing Computer*, Springer, 2007, pp. 3–29.
- [21] N.A. Streitz and R. Wichert, InterLink Deliverable D4.2—Road-Mapping Research in Ambient Computing and Communication Environments, 2009.
- [22] H. Tsujita, K. Tsukada and I. Sioo, SyncDecor: Communication appliances for couples separated by distance, in: *Proc. of Ubicomp*, ACM, 2008, pp. 279–286.
- [23] H. Tsujita, K. Tsukada and I. Sioo, InPhase: A communication system focused on happy coincidences of daily behaviors, in: *Proc. of the 27th Int. Conf. extended abstracts on Human factors in computing systems*, ACM, 2009, pp. 3401–3406.
- [24] G. Vanderheiden and G. Zimmermann, Use of user interface sockets to create naturally evolving intelligent environments, in: *Proc. of the 11th Int. Conf. on Human-Computer Interaction (HCI 2005)*, Lawrence Earlbaum Associates, 2005.
- [25] J. Vermeulen, J. Slenders, K. Luyten, and K. Coninx, Bet you look good on the wall: Making the invisible computer visible, in: *Proc. of Aml 2009*, LNCS, Vol. 5859, Springer, 2009, pp. 196–205.
- [26] G. Zimmermann and G. Vanderheiden, The universal control hub: An open platform for remote user interfaces in the digital home, in: *Human-Comp. Int.*, LNCS, Vol. 4551, Springer, 2007, pp. 1040–1049.
- [27] G. Zimmermann and G. Vanderheiden, A dream... The universal remote console, *ISO Focus+*, Feb, 11–13, 2010. Article available online: trace.wisc.edu/docs/2010-URC-ISOFocus/

- [28] D. Zühlke and N. Thiels, Useware engineering: A methodology for the development of user-friendly interfaces, *Library Hi Tech*, Vol. 26(1), Emerald Group Publishing Ltd., Bingley, UK, 2008, pp. 126–140.
- [29] E.D. Mynatt, J. Rowan, A. Jacobs and S. Craighill, Digital family portraits supporting peace of mind for extended family members, in: *Proc. of CHI 2001*, ACM, 2001, pp. 333–340.
- [30] T. Visser, M. Vastenburg and D. Keyson, SnowGlobe: The development of a prototype awareness system for longitudinal field studies, in: *Proc. of the 8th ACM Conf. on Designing Interactive Systems (DIS'10)*, ACM, 2010, pp. 426–429.
- [31] B. De Ruyter and E. Pelgrim, Ambient assisted-living research in CareLab, *ACM Interactions Journal*, July/August 2007, ACM, 2007, pp. 30–33.