Towards a Framework for the Development of Adaptive Multimodal User Interfaces for Ambient Assisted Living Environments

Marco Blumendorf and Sahin Albayrak

DAI-Labor, TU-Berlin
Ernst-Reuter-Platz 7, D-10587 Berlin, Germany
Firstname.Lastname@DAI-Labor.de

Abstract. In this paper we analyse the requirements and challenges ambient assisted living and smart environments pose on interactive systems. We present a framework for the provisioning of user interfaces for such environments. The framework incorporates model-based user interface development technologies to create a runtime system that manages interaction resources and context information to adapt interaction. This approach allows the creation of adaptive and multimodal interactive ambient assisted living applications.

Keywords: smart environments, multimodal interaction, model-based user interface development, ambient assisted living, multi-access service platform.

1 Introduction

Recent developments show that the information society is evolving into a direction, where computing technology moves from single workstations towards distributed systems of networked interactive devices, appliances and sensors. Such systems provide access to a broad range of services from a variety of application domains and will be installed in different physical locations including homes, offices, cars and public spaces. They facilitate the combination of ambient intelligence with the idea to create applications that are tailored to users, interaction technologies and modalities as well as to the environment and different usage situation. This brings together two trends that, at first glance, seem to follow contrary means. On the one hand side the idea of ubiquitous computing aims at hiding technology and complexity in the environment to provide a more intelligent surrounding. On the other hand the general availability of interactive technology at all times raises the demand for the continuous availability of your personal services at your fingertips. Based on these trends, the idea of smart home environments supporting Ambient Assisted Living (AAL) concepts is currently under heavy development. AAL environments target the assistance of the user during every day live by embedding ambient intelligence into home environments. Aiming at increasing support for everyday activities by technology, smart home environments address any kind of user. However, some user groups are likely to benefit more than other. A main focus is thus currently put on the utilization of technology for elderly and disabled users that can gain support with everyday things they

C. Stephanidis (Ed.): Universal Access in HCI, Part II, HCII 2009, LNCS 5615, pp. 150–159, 2009.

can not master by themselves anymore. In any case, there are numerous issues that need to be solved independently of the targeted user group.

In the following we will elaborate on the technological challenges from a human-computer interaction perspective. We review the current state of the art and identify open issues. Afterwards we introduce a framework for the creation of multimodal user interfaces for smart home environments and identify its main components. A reference implementation in form of the Multi-Access Service Platform, successfully deployed in an ambient assisted living testbed as part of the Service Centric Home project (www.sercho.de), is presented. A conclusion and outlook summarize the work and give an outlook on future work.

2 Challenges for Interaction in AAL Environments

In smart environments the internetworking of resources, including interaction devices (possibly combining multiple interaction resources), sensors and controllable appliances, forms a complex system offering new means for services and interaction. Real-time and reallife issues like continuous availability, extensibility, resource efficiency, safety, security or privacy [15, 2] are major challenges for such systems. However, interactive applications for such environments face additional challenges.

In contrast to PC-based systems, personal devices and applications, interactive systems and applications embedded in the environment need to address multiple users and various user groups with different skills and preferences. Such systems are used in scenarios much less predictable than the usual “user in front of a PC” usage schema [15, 1]. Special needs of different target groups like supportive or rehabilitative usage, non disruptiveness, invisibility, declining capabilities of users, low acceptance for technical problems and the involvement in the active everyday life [1, 19, 2] have to be considered carefully. While personalization puts a strong focus on the user as the main actor for any kind of system, context-of-use adaptivity goes one step further and comprises adaptation to user, platform and environment. Adaptive systems are required to monitor themselves and their environment and provide appropriate reactions to monitored changes before they lead to a disruption of operation [15, 19, 10]. The required context information must be well defined and properly modeled [12]. The complexity of such adaptive systems is massively increased by the distributedness of AAL environments. The availability of multiple resources (interaction, sensors and appliances) raises the need to utilize different resources for interaction, making the adaptation to the different capabilities or even different modalities an essential issue. The interactive systems must be capable of dealing in real time with the distribution of input and output in the environment to provide humans with continuous, flexible, and coherent communication [10]. The distribution of interaction across multiple interaction resources sequentially can provide a richer interaction by addressing the fact that the user moves around in the environment during interaction. Using multiple interaction resources simultaneously takes into account the appropriateness of a combination of resources for a given task over the utilization of a single device. The combination of multiple different interaction resources also leads to the usage of multiple modalities and interaction paradigms. Interaction shifts from an explicit paradigm, in which the user’s attention is on computing, to an implicit paradigm, in which interfaces themselves proactively drive human attention when required [10]. Multimodal interaction can provide greater robustness of the system [16] and natural communication between user and system by voice and gestures can enhance usability [19, 14], especially if keyboard and mouse are not available or suitable to use. However, multimodal interaction is still not entirely understood and interfaces must be carefully designed, respecting each modality's peculiarities, expressiveness, and interaction capabilities [12] and further investigation how users cope with multimodal distributed interaction is required [10]. Distributed multimodal interaction requires the ability to consider an unpredictable number of devices and device federations ranging from mobile phones or headsets to multitouch wall displays and needs to address the lack of a dominating interaction paradigm in AAL environments. Looking at all these challenges, a main factor is the overall interaction experience, which has to excellent, so that users like the vision of being surrounded by of computers (which is usually not the case with today's GUIs) [14]. It is required to establish an appropriate balance between automation and human control [10]. While it is sometimes appreciated if the system learns human behavior patterns, human intervention directing and modifying the behavior of the environment should always be possible [10, 14]. Currently there is a strong lack of development tools, methodologies and runtime environments for such applications [10] and the integration of human-computer interaction and software engineering [14].

In our work we focus on the issues ambient assisted living environments pose to interactive systems at runtime. While development tools and methodologies can address the design issues of the described challenges, there is an urgent need to manage and maintain interaction at runtime. The consideration of context information, adaptation, handling of distribution and processing of multimodal interaction require extensive efforts at runtime. After introducing the state of the art in the next section, we thus focus on deriving an architecture addressing the identified issues at runtime.

3 Related Work

Recent work in the area of ambient assisted living, ambient intelligence and smart home environments strive for the development of a common architecture and framework for AAL environments [11, 12, 2]. A main aspect of such frameworks is the integration of home control, sensor information and interaction means. The development and runtime management of multimodal interaction can be greatly supported by model-based approaches. Utilizing formal user interface models takes the design process to a computer-processable level that makes design decisions understandable for automatic systems. While earlier work in this field addresses the derivation of multiple consistent user interfaces from a single user interface model at design time, the focus has recently been extended to the utilization of user interface models at runtime [4, 6, 8, 18]. This provides the possibility to exploit the information contained in the model for user interface adaptation and the handling of multimodal interaction within changing contexts-of-use. Coninx [6] presents a runtime system that targets the creation of context-aware user interfaces. Similarly the DynaMo-AID Runtime Architecture [5] processes context information and utilizes user interface models to build adaptive user interfaces. Tycoon [13], SmartKom [17] as well as the conceptual structure of a multimodal dialogue system [7] and the Framework for Adaptive Multimodal Environments (FAME) [9] present architectures for multimodal interaction processing at runtime. Runtime issues like the coordination and integration of several communication channels that operate in parallel (fusion), the partition of information for the generation of efficient multimodal presentations (fission), and the synchronization of the input and output and feedback as well as adaptation to the context of use can be supported by the utilization of user interface models. In the following we present a framework, utilizing model-based technology for the creation of interactive AAL systems. It addresses the presented challenges and aims at managing multimodal user interfaces for AAL environments.

4 A Model-Based Framework for AAL Services

From a user interface perspective a framework addressing the identified challenges has to support adaptation, distribution, migration and multimodality of user interfaces. Reflecting the clash of user interfaces for AAL environments and the ubiquitous computing paradigms, we propose to call such user interfaces Ubiquitous User Interfaces (UUIs). Recent work in this area has shown that model-based approaches can help addressing the various issues. In the following we identify the core features of a framework, combining UI models with a runtime system supporting UUIs.

Figure 1 shows the general architecture. Based on a model of the user interface the architecture aggregates context information, fusion, fission and adaptation components as well as a connection to backend-services on the one and communication channels to the interaction resources on the other hand. A Dialog Manager handles the dialog based on the set of models, representing the interaction and user interface design on different levels of abstraction. For each interaction step it calculates the current dialog state based on the model, recent interaction and relevant context information. Context information is provided through a comprehensive context model, reflecting information perceived by observers (sensors) as well as predefined information like user preferences. It also maintains the available interaction resources based on which the distribution component distributes the current dialog state. Once the relation of interaction elements and interaction resources has been calculated, the information is passed to the Interaction Manager. It then takes care of the creation and management of the physical user interface from the internal state of the interactive system and the delivery of the different UI parts via communication channels. Communication channels connect interaction resources directly to the runtime system and make them accessible for interaction. They abstract from the underlying communication technology (e.g. HTTP) and deliver final user interface and continuous updates to the user and user input back to the system. Each UI part is thereby build up to a full user interface (e.g. a webpage for a browser) representing the given part of the interaction. User input received through a channel as result of an interaction is matched by the interaction manager and passed to the fusion component. The fusion component aims at interpreting the user input based on the expected input and relates input from different resources and modalities to find a suitable interpretation. The found interpretation is then handed to the dialog manager as a basis to calculate the next dialog state based on the models and the different levels of abstraction. Additionally, a call to the application logic can be issued via the service model to trigger any system action. Backend services also have the possibility to access context information and to directly alter the UI presentation, e.g. to proactively contact the user or present crucial notifications. An API allows external access to directly control the user interface via the adaptation component. Once the new UI state has been calculated it is again distributed and the perceivable presentations are updated. The distribution component is also responsible to trigger the migration of (part of) the UI to alter the distribution even within a dialog step.

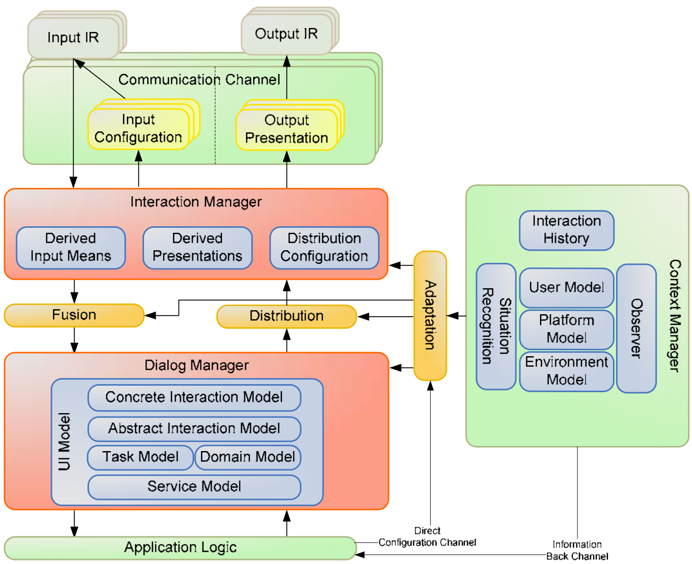

Fig. 1. A runtime architecture, reflecting the identified requirements

The described approach allows the connection of multiple interaction resources to a server-side system managing the interaction and UI distribution. The architecture separates input and output IRs, to be able to (dynamically) incorporate additional IRs at any time (top of Figure 1). While output IRs are provided with a presentation UI, input IRs can be configured i.e. to limit the possible input. This approach e.g. allows the limitation of the vocabulary of a speech recognizer or the incorporation of a gesture device and the restriction of the known gestures at runtime. Input processing as well as the response generation happen based on a multi-step process, enriching received input and created output with additional means on different levels of the architecture and the UI model. Based on this reference architecture, we developed the Multi-Access Service Platform, a model-based runtime system that allows the derivation and adaptation of user interfaces from design models at runtime. In the following we illustrate selected aspects of the platform.

5 The Multi-Access Service Platform

The described architecture has been implemented as the Multi-Access Service Platform (MASP) in the Service Centric Home Project. The MASP aims at the creation of ubiquitous user interfaces for smart home environments and provides a model-based runtime system to create distributed multimodal interaction. The core of the MASP is a set of user interface models representing the interaction on different levels of abstraction. The well-accepted Cameleon Reference Framework [4] proposes four level of abstraction (task & domain, abstract, concrete & final UI) which has been incorporated into the framework above. Figure 2 shows an overview of the net of models we are currently utilizing in the MASP. It follows the same abstraction levels, but also introduces additional models for configuration purposes. The models are connected by mappings and executed at runtime as we described in [3]. This allows the models to dynamically evolve over time and describe the application, its current state and the required interaction as a whole instead of providing a static snapshot only.

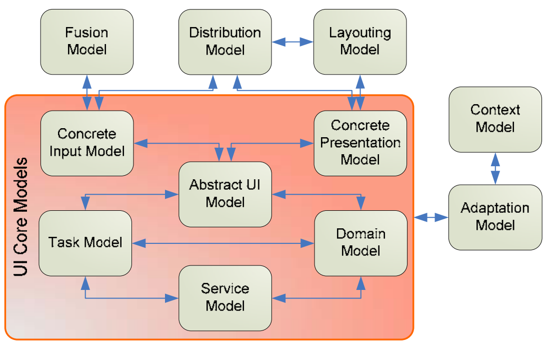

Fig. 2. The MASP user interface models, maintained by the dialog manager and the mappings between the different parts

While the service model connects a given backend system at runtime, the task- and domain-model describe the basic concepts underlying the user interface. Based on the defined concepts, the interaction models define the actual communication with the user. They thereby define abstract interaction elements (Abstract Interaction Model), aiming at a modality and device independent description of interaction and concrete elements (Concrete Input & Concrete Output Model), targeting specific modalities and devices. The interaction model has a main role to provide support for multimodal interaction. A context model provides interaction-relevant context information. The model holds information about the available interaction resources and allows their incorporation into the interaction process at runtime. Additionally it provides information about user and environment and comprises context providers, continuously delivering context information at runtime. The model is continuously updated at runtime to reflect the current interaction context. Based on these models a fusion and a distribution model define fusion and distribution rules to support multimodal and distributed user interfaces. A layouting model produces layout constraints for different usage scenarios. Finally a mapping model interconnects the different models and ensures synchronization and information exchange between them. By linking the task model to service and interaction model, the execution of the task model triggers the activation of service calls and interaction elements. While service calls activate backend functions, active interaction elements are displayed on the screen and allow user interaction. They also incorporate domain model elements in their presentation and allow their manipulation through user input as defined by the mappings. The context model influences the presentation of the interaction elements that are related to context information. Thus, the execution of the task model triggers a chain reaction, leading to the derivation of an interaction state from the defined user interface model. This state contains a set of concrete interaction elements (as well as the related tasks and abstract interaction elements) that are the basis for the distribution calculation. Based on the distribution rules, the elements are assigned to interaction resources and layout rules are applied for each resource. The delivery of the elements and the final rendering are done by the interaction manager and the communication channels. The described models are created by the application developer, based on a set of metamodels that outline the modeling possibilities. Each metamodel can also define adaptation possibilities (construction elements) that can be applied at runtime. These can be used by the designer to create an adaptation model for his application influencing the final presentation of the user interface based on context information.

Fig. 3. Screenshot of our implementation of the meta-UI

Based on these models, the MASP provides a comprehensive set of features that can be directly controlled via the Meta-UI shown in Figure 3. Additionally the features can also be incorporated into applications to support application specific behavior. This includes control over used modalities, the migration of the user interface by directly choosing interaction resources, the distribution of parts of the user interfaces, and the (de-)activation of predefined adaptation means. While our underlying models aim at describing the anticipated interaction, the components of the architecture provide the features at runtime. Multimodality is realized by the distribution component, segmenting the user interface across multiple resources supporting multiple modalities, and the fusion component, matching and interpreting input from different modalities. Both components are strongly related to the underlying user interface model allowing the configuration of the components. The distribution component takes account of the information about all available interaction resources, stored in the context model to calculate the optimal usage of the available interaction resources based on the current status of the interaction of the dialog manager. Using the active input elements from the concrete input model and the active output elements from the concrete presentation model it calculates a distribution, also taking into account the relations between the elements expressed on the other abstraction levels. Interaction elements related to the same task are e.g. likely not to be separated. A main goal of the distribution is to support as many different input capabilities as possible and guidance to the user about how to use these. Additionally the recently used interaction resources and modalities are considered to ensure consistency during the interaction.

Once the distribution has been calculated, the assigned user interface elements are delivered via the communication channels, which also perform resource specific adaptations of the final user interface and handle the communication. Utilizing these channels ensures full control over each used interaction resource by providing the means to manipulate the user interface. In terms of output i.e. adding and removing presentation elements at any time, in terms of input i.e. altering the input configuration. Input configurations help restricting the possible user interaction in a single modality and thus allow incorporating context and influencing the recognition engines. In contrast to other multimodal approaches we did not put a main focus on the semantic analysis of any user input but at the provisioning of a multimodal user interface guiding and restricting the possible interaction. As input configurations are derived from the abstract and concrete interaction models, these models are also used to interpret the received input and derive its meaning. While the abstract interaction model combines input and output and focuses on the modality independent definition of the interaction goals in terms of commands, choices, free input and output, the related concrete interaction model separates input and output interfaces and aims at the definition of concrete interaction necessary to complete the abstract interaction goals.

After preprocessing and filtering monomodal input by the channel and the interaction manager, the fusion engine combines multiple inputs based on information from the concrete interaction model and its configuration in the fusion model. Afterwards this input is evaluated in terms of the goals of the abstract interaction model and then finally brought down to the task and domain level where it either alters a domain object or completes a task (which could also lead to a service invocation). Each interaction can thus be interpreted on multiple levels of abstraction before finally proceeding to the next interaction step of the dialog. The combination of multiple devices allows combining complementary devices to enhance the interaction and support multiple modalities and interaction styles. A common example for this feature is the utilization of a mobile device as a remote control for a large display by distributing an application across devices and modalities.

Finally this whole user interface derivation and input interpretation process can be influenced by an adaptation component. The component is configured by an adaptation model, holding adaptation rules, and has access to all models. Similar to the evolution model of [18] the adaptation model defines an adaptation as graph transformation, denoting the node(s) to apply the transformations to (left-hand-side) and a description of the alteration of these nodes (right-hand-side). The alteration of the nodes is defined in form of construction elements that are provided by each metamodel to predefine possible adaptations. Adaptations are triggered by context situations and thus by states reached by the context model. Due to the availability of design information in the user interface models at runtime, the defined adaptations can address a broad spectrum of model alterations. However, in contrast to other approaches, the possible adaptations for each model are determined by its metamodel and thus only well defined adaptation can be performed and the integrity of the models is ensured at any time.

6 Summary and Outlook

In this work we presented a framework for the creation of smart home user interfaces, supporting context-sensitive, multimodal interaction. The framework aims at providing application spanning functionality to ease the development of future smart home applications. User interface adaptation, migration, distribution and multimodal interaction have been in the focus for the development of the framework. Based on the developed framework we created the Multi-Access Service Platform (MASP, masp.dai-labor.de), a reference implementation using executable user interface models to derive distributed multimodal user interfaces. The MASP has been successfully used in the Service Centric Home Project and different multimodal applications have been built. This includes a cooking assistant, an energy assistant and a meta user interface allowing to control the application-spanning features (adaptation, migration, distribution and multimodality). We are currently creating a health monitor example applications, providing ubiquitous access to your health data, based on a portable sensor platform and MASP supported interaction.

In our future work we aim at broadening our approach to other surroundings like offices, public spaces, cars, or other mobile situations. This also leads to the concept of personal interactive spaces, virtually surrounding a user continuously. Additionally the improvement of the input processing capabilities and the multimodal interaction are a major focus as well as the improvement of existing and the development of additional adaptation means. Based on our Ambient Assisted Living Testbed we also want to conduct further user studies and put a stronger focus on elderly and disabled users as main target groups.

References

1. Abascal, J., Fernández de Castro, I., Lafuente, A.L., Cia, J.M.: Adaptive interfaces for supportive ambient intelligence environments. In: Miesenberger, K., Klaus, J., Zagler, W.L., Karshmer, A.I. (eds.) ICCHP 2008. LNCS, vol. 5105, pp. 30–37. Springer, Heidelberg (2008)
2. Becker, M.: Software architecture trends and promising technology for ambient assisted living systems. In: Assisted Living Systems - Models, Architectures and Engineering Approaches. Dagstuhl Seminar Proceedings, vol. 07462 (2008)
3. Blumendorf, M., Lehmann, G., Feuerstack, S., Albayrak, S.: Executable models for human-computer interaction. In: Interactive Systems - Design, Springer, Heidelberg (2008)

4. Calvary, G., Coutaz, J., Thevenin, D., Limbourg, Q., Souchon, N., Bouillon, L., Florins, M., Vanderdonckt, J.: Plasticity of user interfaces: A revised reference framework. In: Proceedings of the First International Workshop on Task Models and Diagrams for User Interface Design, pp. 127–134. INFOREC Publishing House (2002)
5. Clerckx, T., Vandervelpen, C., Luyten, K., Coninx, K.: A task-driven user interface architecture for ambient intelligent environments. In: Proceedings of the 11th international conference on Intelligent user interfaces, pp. 309–311. ACM Press, New York (2006)
6. Coninx, K., Luyten, K., Vandervelpen, C., Van den Bergh, J., Creemers, B.: Dygimes: Dynamically generating interfaces for mobile computing devices and embedded systems. In: Chittaro, L. (ed.) Mobile HCI 2003. LNCS, vol. 2795, pp. 256–270. Springer, Heidelberg (2003)
7. Delgado, R., Araki, M.: Spoken, Multilingual and Multimodal Dialogue Systems. John Wiley & Sons, Ltd., Chichester (2006)
8. Demeure, A., Calvary, G., Coutaz, J., Vanderdonckt, J.: The comets inspector: Towards run time plasticity control based on a sematic network. In: Proceedings of the 5th annual conference on Task models and diagrams (2006)
9. Duarte, C., Carriço, L.: A conceptual framework for developing adaptive multimodal applications. In: Proceedings of the 11th international conference on Intelligent user interfaces, pp. 132–139. ACM Press, New York (2006)
10. Emiliani, P.-L., Stephanidis, C.: Universal access to ambient intelligence environments: opportunities and challenges for people with disabilities. IBM Syst. J. 44(3), 605–619 (2005)
11. Kirste, T., Rapp, S.: Architecture for multimodal interactive assistant systems. In: Statustagung der Leitprojekte "Mensch-Technik-Interaktion" (2001)
12. Emiliani, P.-L., Burzagli, L., Gabbanini, F.: Ambient intelligence and multimodality. In: Stephanidis, C. (ed.) UAHCI 2007 (Part II). LNCS, vol. 4555, pp. 33–42. Springer, Heidelberg (2007)
13. Martin, J.-C.: Tycoon: Theoretical framework and software tools for multimodal interfaces. In: Intelligence and Multimodality in Multimedia interfaces. AAAI Press, Menlo Park (1998)
14. Mühlhäuser, M.: Multimodal interaction for ambient assisted living (aal). In: Assisted Living Systems - Models, Architectures and Engineering Approaches. Dagstuhl Seminar Proceedings, vol. 07462 (2007)
15. Nehmer, J., Becker, M., Karshmer, A., Lamm, R.: Living assistance systems: an ambient intelligence approach. In: Proceedings of the 28th international conference on Software engineering, pp. 43–50. ACM, New York (2006)
16. Reeves, L.-M., Lai, J., Larson, J.-A., Oviatt, S., Balaji, T.-S., Buisine, S., Collings, P., Cohen, P., Kraal, B., Martin, J.-C., McTear, M., Raman, T.-V., Stanney, K.-M., Su, H., Wang, Q.-Y.: Guidelines for multimodal user interface design. Commun. ACM 47(1), 57–59 (2004)
17. Reithinger, N., Alexandersson, J., Becker, T., Blocher, A., Engel, R., Löckelt, M., Müller, J., Pfleger, N., Poller, P., Streit, M., Tschernomas, V.: Smartkom: adaptive and flexible multimodal access to multiple applications. In: Proceedings of the 5th international conference on Multimodal interfaces, pp. 101–108. ACM Press, New York (2003)
18. Sottet, J.-S., Ganneau, V., Calvary, G., Coutaz, J., Demeure, A., Favre, J.-M., Demumieux, R.: Model-driven adaptation for plastic user interfaces. In: Baranauskas, C., Palanque, P., Abascal, J., Barbosa, S.D.J. (eds.) INTERACT 2007. LNCS, vol. 4662, pp. 397–410. Springer, Heidelberg (2007)
19. Weber, W., Rabaey, J.-M., Aarts, E. (eds.): Ambient Intelligence. Springer, Heidelberg (2005)