End-User Composition Interfaces for Smart Environments: A Preliminary Study of Usability Factors

Yngve Dahl[1] and Reidar-Martin Svendsen[2]

[1]SINTEF ICT,
7465 Trondheim, Norway
[2] Telenor Corporate Development,
7004 Trondheim, Norway
yngve.dahl@sintef.no, Reidar-Martin.Svendsen@telenor.com

Abstract. This paper describes a preliminary study of factors that influence the usability of end-user composition interfaces for smart environments. Three early GUI prototypes were tested in a usability laboratory, and transcriptions from the test subjects' comments during the experiment were analyzed in search of recurring areas of concern. Four usability factors were identified: (1) predictability of composition model, (2) readability of composition representation, (3) overview and means for planning compositions, and (4) attractiveness and desirability.

Keywords: End-user composition, Graphical user interfaces, Interface Metaphor, Smart environments, Usability.

1 Introduction

The realization of smart environments is often framed as a technical problem related to the lack of mechanisms that allow heterogeneous devices to exchange data and respond to local events [1]. Realizing smart environments, however, also requires intelligible user interfaces that allow people to customize and control their local surroundings. This paper describes a preliminary comparative study of graphical user interfaces (GUIs) designed to help end-users combine networked resources inside a smart environment into compositions that can offer new extended functionality. The focus is on GUIs that are applicable for design-time composition, that is, specification of compositions in advance of using them. The evaluated GUIs primarily target composition editors running on desktop computers.

The main objective of this paper is to identify factors influencing a composition interface's effectiveness, efficiency, and satisfaction in use. Effectiveness, efficiency, and user satisfaction are key elements in the ISO definition of usability [2]. We refer to factors affecting these three elements as *usability factors*.

Three composition interfaces were prototyped and tested as part of a problem solving exercise. Test subjects were given tasks related to composition of devices and services provided in hypothetical domestic smart environment. The experiment was conducted in a usability laboratory with sixteen test participants (architect and IT students). Transcriptions from the users' comments during testing were analyzed to

A. Marcus (Ed.): Design, User Experience, and Usability, Pt II, HCII 2011, LNCS 6770, pp. 118–127, 2011.

identify factors that influenced the usability of the composition interfaces, as perceived b the test subjects.

The paper's main contribution is a qualitative understanding of how the usability of end-user composition interfaces is intimately dependent a set of critical factors.

2 Related Work

Alternative approaches to end-user composition have been explored in earlier work. Humble et al. [3] describe an editor that allows users to configure a domestic ubiquitous computing environment. The GUI of the editor is based on a jigsaw metaphor. Users can combine functionalities offered by various components, by coupling graphical puzzle-like pieces that represent the components. Newman [4] proposes a concept where "cooking recipes" form the underlying metaphor. The digital recipes describe the required ingredients (hardware and software components) and preparations (steps) to make a composition. The Speakeasy system [10] employs a web browser interface and task-oriented templates. The Device Composition Aquarium [5] displays compositions as graphs or "wires".

The studies cited above illustrate some of the variety of design metaphors and concepts that have been investigated in the context of end-user composition. Although various approaches have been proposed, there is relatively little practical guidance as to what interface characteristics that influence the usability of various approaches.

3 Overall Research Design

The conducted evaluation is formative in nature. As opposed to measuring user performance (e.g., task solving time, number of slips and breakdowns, etc.) for different user interfaces, our goal was to evoke usability-related reflections among test subjects to help inform further design. We have observed architect and IT students interacting with early prototypes to solve composition tasks, and asked them to articulate their experience from use. This has resulted primarily in qualitative data. Given the limited expressive power of the prototypes (see Sect. 4.1), we chose not to investigate possible correlations between programming experience and preference for certain composition interfaces. Instead we focus exclusively on the qualities and drawbacks of the prototypes as articulated by the test subjects.

The evaluation was conducted in a usability laboratory. The test subjects used a conventional PC with keyboard and mouse to test the composition interfaces. Earlier work [6] has identified PCs as the user preferred tool for specifying automation of tasks that can be predetermined and planned.

To give test participants an initial idea of the basic purposes of the prototypes they were given a brief introduction to a hypothetical domestic smart environment. They were asked to imagine that they were living in a resident where various domestic devices and sensors could offer services, respond to local events, and exchange data over a digital network. To help concretize the setting, the test subjects were shown a paper sheet illustrating the different types of devices, or components, they would be dealing with as part of the composition tasks (Fig. 1). The overview also indicated the assumed physical location of the various components.

Fig. 1. The various components that the test participants would be dealing with as part of the composition tasks were illustrated on a paper sheet

4 Prototypes and Experimental Setup

4.1 Simplicity in Use vs. Expressive Power

In order to conduct a comparative evaluation of end-user composition interfaces it is necessary that all candidate solutions allow the same behavior to be specified. This, again, raises the need for a common composition model.

When developing tools for specifying system behavior *simplicity in use* and *expressive power* form two mutually exclusive criteria [3]. Conventional programming languages maximize expressiveness, but raise the threshold for successful use in doing so.

Given our focus on evaluating early concepts, we chose to limit the expressive power of the prototypes to simple condition-response rules. For example, a composition could specify that if an in-house smoke detector is triggered, an SMS notification should be sent to the house owner's mobile phone. In this case, the execution of a service associated with one device (due to some external event), causes another predefined device to respond by executing another service.

For simplicity, we also decided to omit features such as specification of parameters (e.g., mobile phone number, SMS content) related to different devices and services.

4.2 Prototypes

Inspired by earlier concepts (e.g., [3, 5]), three prototypical composition interfaces were build. The three prototypes were *filtered lists, wiring diagram*, and *jigsaw puzzle*. Below, we describe each prototype in further detail.

Filtered Lists Prototype. This prototype allows users to specify a composition through sequential steps. A composition is defined by subsequently specifying the condition (component and event) and the response (component and service) from respective lists (Fig. 2). Filters are applied so that the lists only present alternatives that are relevant for previous selections. For examples, if a user first selects a smoke detector as the triggering device, only events that the smoke detector can respond to are shown in the events list. Relevant response components and services are only shown after the trigger device and event has been selected.

Saved compositions are shown in a dedicated table.

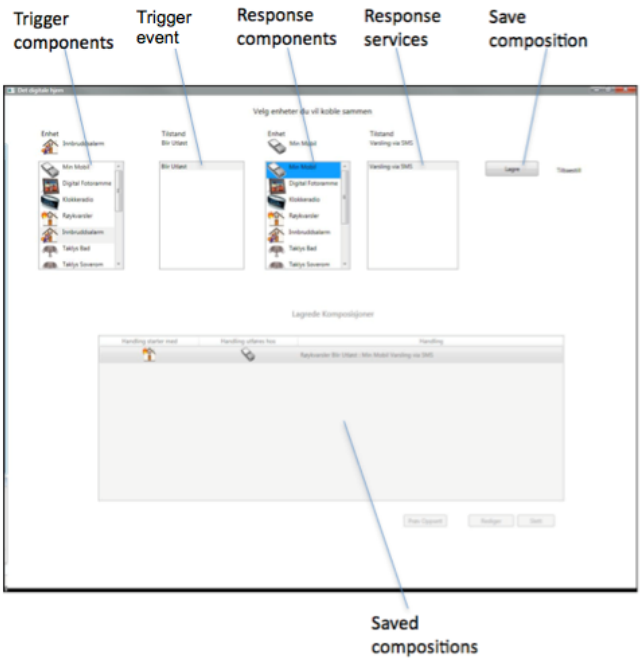


Fig. 2. Filtered lists prototype

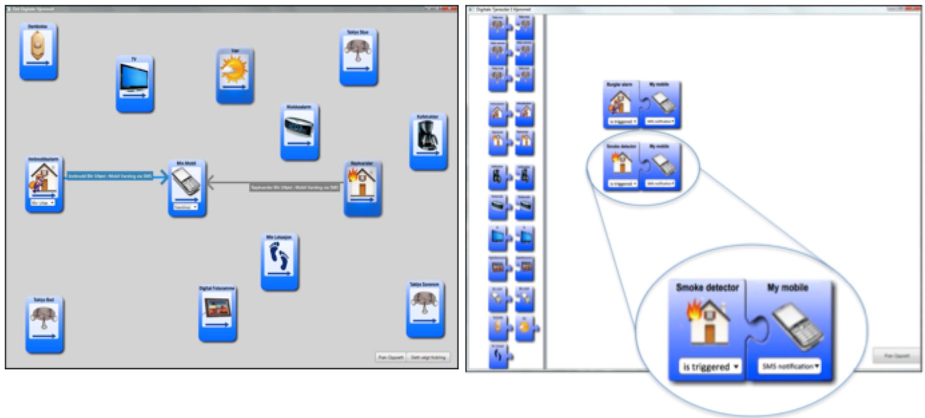
Wiring Diagram Prototype. The metaphor of coupling or "wiring" together separate GUI elements to denote relationships, has been extensively used in different types of web composition tools (e.g., Microsoft Popfly). The metaphor builds on the way stereo system components (tuners, amplifiers and speakers) are hardwired together.

The wiring prototype (Fig. 3a) was implemented as a more direct manipulation approach to end-user composition, compared to that of the filtered lists. Draggable nodes contained within a composition workspace represent the different components. Each node shows a symbol and the name of the component it represents.

To form a composition, a user first selects the trigger node and then drags a graphical arrow or "wire" to the intended responding node. The graphical arrows indicate the direction of control flow.

After two nodes have been connected, the user can specify the respective events and services that are to be part of the composition via dropdown lists contained within the nodes.

Existing compositions can be modified by rearranging the connecting arrows between nodes.



(a) (b)

Fig. 3. (a) Wiring diagram prototype. (b) Jigsaw puzzle prototype (with close-up)

Jigsaw Puzzle Prototype. The jigsaw puzzle metaphor is the alternative among the evaluated alternatives with the most direct mapping to a real-world object. Similar to the wiring prototype, this prototype also draws on the principle of direct manipulation. Users specify compositions by "snapping" together (dragging and dropping) two interlocking puzzle-like GUI elements representing various components (Fig. 3b).

Puzzle pieces with knobs represent trigger components, while puzzle pieces with holes represent response components. Similar to the wiring diagram, the trigger and response services are specified via dropdown lists.

For each composition involving a specific component, a new puzzle piece representing that component has to be collected from the "tray" (left), and dragged onto the composition workspace (right). Thus, the workspace can contain multiple pieces representing the same component (one per composition). This is in contrast to the wiring prototype, in which a single node is used for all compositions the corresponding component is to be part of.

Compositions can be deleted by separating two attached pieces.

4.3 Experimental Setup

Test setting and Equipment. The experiment was conducted in a usability laboratory equipped with high fidelity video and audio recording equipment. Screen mirroring software allowed us to record video of GUI interaction during the trials. A conventional PC with keyboard and mouse were used in the testing of the GUI prototypes. The prototypes were tested in random order.

To allow test participants to validate the run-time effect of compositions a software simulation tool was used. The simulation tool could give a simple visual representation of which and how components were affected given a specified condition.

Composition Tasks. The test participants were given a set of seven composition tasks (Fig. 4) to solve using the prototypes.

1. I want to be notified quickly in case of possible burglary or fire in my home.
2. The lighting in my bedroom should be on when I am there and off when I am not there.
3. I often take photos with my mobile phone when I am on travel. It would be nice if photos I am sending to my home could be displayed on the wall in the living room.
4. I am often in a hurry in the morning before I leave for work, but I usually need a cup of coffee to wake myself up. It would have been helpful if I could wake up to freshly brewed coffee.
5. I often wonder what the weather is like when I awake in the morning on working days.
6. When I am watching TV at home, I do not always hear if the doorbell rings.
7. I regret that I asked the house to brew coffee when I wake in the morning.

Fig. 4. Composition tasks

To avoid that the formulation of the composition tasks led to preference bias toward certain interfaces, tasks were presented as problems as opposed to instructions explicitly informing test subject what to compose. Thus, for many tasks there were multiple solutions. The motivation behind the problem-based tasks was to encourage interaction with the prototypes and to evoke creativity among participants.

5 Results and Analysis

5.1 Results

Filtered Lists. Nearly all test subjects stated that they experienced the filtered lists concept as a highly efficient way of specifying compositions. Although we did not measure task execution time, video recordings from the testing indicate that participants managed to come up with solutions to the composition tasks fairly rapidly compared to the other prototypes. Test subjects expressed that the interface provided a good overview over the different components and their associated services, and that it felt like a (quote) “safe”, “confident” and “logical” solution.

Many test subjects found that the orderly presentation of the components and services helped give them a quick idea of how to specify compositions. It was also pointed out that the GUI made it made easy to see alternative solutions to composition tasks.

Wiring Diagram. The filtered lists prototype gave explicit indications to users which screen elements that were intended for denoting the trigger part, and which screen objects that were intended for denoting the response part of a composition. The different parts were specified sequentially through lists organized in a left-to-right fashion. In the wiring diagram prototype, however, the issue of control flow between trigger and response component was not as apparent. Here, the users had to specify control flow by means of the connecting arrows. We noticed that participants, particularly those with no programming background, often did not pay attention to control flow when using the wiring diagram prototype. Their comments suggested that they did not think of the arrows as a way of specifying control flow, but rather a way of specifying two objects that should exchange data.

There were mixed opinions regarding how appropriate the wiring diagram prototype served as a composition interface. Some participants expressed that it was difficult to visually "trace" the connecting arrow to see which components it linked. Test subjects also expressed that as the number of compositions increased, the connecting arrows would clutter the interface, making it difficult to get an overview of existing compositions. One user commented that saved compositions could preferably be listed in a tabular form, as in the filtered lists prototype. Concerns related to overview of compositions arose primarily in the planning of new compositions. In this process, we found that test subjects frequently reviewed existing compositions to evaluate the how new composition would fit into the collection of existing ones.

There were also general concerns that the current approach would require users to frequently tidy up the user interface to get a clear overview of specified compositions. While the lack of overview was seen as a challenge in the process of specifying compositions, some test participants found that the presentation could be suitable for visualizing relationship between devices, for example, how many compositions a particular device is involved in.

Jigsaw Puzzle. The jigsaw puzzle prototype was generally viewed as an intuitive and fun way of building compositions. Participants commented that the interlocking pieces made it easy to see which devices that were part of a composition. The ability to provide this type of information clearly was one of the main challenges associated with the wiring diagram, where users often found themselves "tracing wires".

At the same time, it was frequently pointed out that the jigsaw puzzle prototype, similar to the case with the wiring diagram prototype, tended to gradually become disorderly and difficult to extract information from as the number of compositions increased. The number compositions that were specified during the task solving (typically between eight to ten) was, according to several test subjects, close to the limit of composition that could be visualized in the current GUI without significantly compromising the general overview.

We observed that some test subjects found it challenging to differ between trigger and response device when using the jigsaw puzzle prototype. After being presented a task, they would often put together pieces representing relevant components, but not pay attention to which piece that formed the trigger part and which piece that formed the response part. Similar issues related to control flow also applied to the wiring diagram prototype.

Test participants also gave indications that it was challenging to quickly understand and extract meaning out of compositions represented as jigsaws. Similar remarks related to difficulties in interpreting graphical representations of compositions quickly were also given in the testing of the wiring diagram prototype.

Test participants generally expressed that they enjoyed the direct manipulation of screen objects that the jigsaw prototype allowed for. The video recordings also indicate that the prototype encouraged users to try out alternative solutions to the composition tasks.

5.2 Analysis

Based on an investigation of the test subjects' transcribed responses, we identified four factors, or reoccurring areas of concern, that strongly affect the usability of the composition interfaces. These were (1) *predictability of composition model*, (2) *readability of composition representation,* (3) *overview and means for planning compositions*, and (4) *attractiveness and desirability*.

Predictability of Composition Model. The composition model used in the current study was equivalent to condition-response rules. While the three GUI prototypes all matched this model, comments form the test subjects and their interaction with the prototypes, suggested that the extent to which each solution effectively communicated the underlying composition model varied.

The user-perceived predictability (or lack thereof) of the composition model can in many ways be seen as a result of the way GUI elements for specifying compositions were arranged in the different prototypes. In the filtered lists prototype, the fixed left-to-right arrangement of GUI elements for specifying composition aligned with the format of condition-action rules. This was not the case for the other two prototypes. The nodes representing each device in the wiring diagram were initially distributed across the composition workspace. The jigsaw pieces were initially placed in the "tray" beside the composition workspace.

The user-perceived predictability of the composition model can also be seen in light of control issues that test participants experienced. Because the filtered lists prototype required the condition and response part to be specified in a sequential manner users avoided problems related to specification of control flow, which was sometimes the case with the other prototypes. With respect to the wiring diagram and jigsaw puzzle prototypes, control flow had to be handled explicitly by the users. For the wiring prototype this meant pointing the connecting arrows between nodes in the "right" direction. For the jigsaw puzzle control flow was specified in a left-to-right fashion (knobs and holes).

The aspects pointed out above, concerning arrangement of GUI elements and control flow, explain why test subjects experienced a feeling of greater control over the filtered lists prototype vis-à-vis the others.

Readability of Composition Representation. The usability of end-user composition interfaces does not only depend on their ability to provide intelligible interfaces for combining different components. We also found that test subjects frequently took existing compositions represented in the GUI in consideration when planning which devices and services that should be part of new compositions. This makes it essential that the user interface also effectively communicates the overall picture of compositions. One important lesson learned from the experiment is that what makes a potentially suitable metaphor for conveying to users how to manipulate GUI elements to form compositions, might in some cases compromise composition readability.

Our findings related to the wiring diagram and jigsaw puzzle prototype exemplifies how certain notations can have negative effects on readability of compositions. The representation of compositions in these two prototypes required more cognitive effort to "decode" and understand the run-time effect. In the case of the wiring diagram prototype this decoding meant that users had to trail wires to identify associated components, and also pay attention to pointing direction of the attached arrow. With regard to the jigsaw puzzle prototype, test subjects found themselves mentally "decomposing" puzzles.

Overview and Means for Planning Compositions. Another issue which surfaced as test participants were in the process of forming new compositions, relates to the means for planning compositions ahead. As described earlier, the wiring diagram prototype and the jigsaw puzzle prototype did not explicitly distinguish between the composition workspace and the spaces providing overview of existing compositions.. A reoccurring concern among test participants regarding these solutions was that the GUIs ability to effectively present a comprehensible overview of compositions was reduced as the number of compositions increased. In a sense, the user interface metaphor and the graphical elements used to represent compositions (wires and jigsaw puzzle pieces) would at a certain point have a negative effect on the overall overview.

We consider the issues concerning readability and overview to be particularly relevant for design of composition editors that allow for more complex compositions. Challenges related to, e.g., cognitive efforts involved in extracting meaning from composition representations could easily escalate in such cases.

Attractiveness and Desirability. The last factor, which we identified as being central for the user-perceived usability of composition techniques, relates to the overall experience of use. In both the wiring prototype and the jigsaw puzzle prototype compositions were formed by means of direct manipulation (drag and drop) of interface elements. Judging from the feedback from test participants, they experienced these solutions, particularly the latter, as being more playful and engaging compared to the filtered lists prototype. The reoccurring argument among test participants with a preference for the jigsaw puzzle prototype was that is offered a fun way of forming compositions. As described earlier, this approach also appeared to encourage exploration of alternative solutions to the tasks.

While most test subjects experienced the filtered lists prototype as being the most efficient in use among the alternatives, responses such as those described above indicate that designing appealing compositing interfaces is important for user acceptance. In more recent HCI literature, it has been argued that aesthetics and desirability in design are just as important as usability [7].

We are aware that the "fun factor" of the jigsaw prototype might have been given different priority depending on the consequences of making undesirable compositions.

This study was conducted in a laboratory setting and run-time effects of compositions were only simulated. In a real use environment, particularly in the cases related to home security, confidence in use might be given higher priority.

6 Concluding Remarks

We consider the identified usability factors to be essential parts of the "value set" against which end-user appraise composition interfaces. As such, we recommend that designers are mindful about these factors during design.

Acknowledgements. This work was funded in part by the ISIS project (Norwegian Research Council grant # 180122) and a joint research collaboration between SINTEF ICT and Telenor ASA. Thanks to MSc. Bjarne Muri for contributing to prototyping and usability experiments.

References

1. Maestre, J.M., Camacho, E.F.: Smart home interoperability: the DomoEsi project approach. International Journal of Smart Home 3(3) (2009)
2. ISO 9241-210: Ergonomics of human-system interaction – Part 210: Human- centred design for interactive systems. International Organization for Standardization (2010)
3. Humble, J., et al.: "Playing with the bits" user-configuration of ubiquitous domestic environments. In: Dey, A.K., Schmidt, A., McCarthy, J.F. (eds.) UbiComp 2003. LNCS, vol. 2864, pp. 256–263. Springer, Heidelberg (2003)
4. Newman, M.W.: Now We're Cooking: Recipes for End-User Service Composition in the Digital Home. In: CHI 2006 Workshop: IT@Home: Unraveling Complexities of Networked Devices in the Home (2006)
5. Sohn, T., Dey, A.: iCAP: an informal tool for interactive prototyping of context-aware applications. In: CHI 2003 Extended Abstracts on Human Factors in Computing Systems, Ft. Lauderdale, Florida, USA. ACM, New York (2003)
6. Koskela, T., Väänänen-Vainio-Mattila, K.: Evolution towards smart home environments: empirical evaluation of three user interfaces. Personal and Ubiquitous Computing 8(3-4), 234–240 (2004)
7. Norman, D.A.: Emotional Design: Why We Love (Or Hate) Everyday Things. Basic Books, New York (2004)