# Solving One Task with Many Devices

***Marta Kristín Lárusdóttir***
Reykjavik University
Reykjavik, Iceland
marta@ru.is

***Ebba Thora Hvannberg***
University of Iceland
Reykjavik, Iceland
ebba@hi.is

## Abstract

The complex world in which a user can accomplish the same task with many different devices puts many demands on user interface models. This presentation describes such a project where a user-centered approach was used. Models were used to describe users, tasks, user environments and the user interface for different mobile and desk devices. Since tasks require users to go between devices, a navigational model and a more abstract architectural model were developed. Additionally the preparation of usability tests with many devices and environments are described.

## Introduction

People can solve the same task by using many different devices in different environments. For keeping track of your calendar you can use a calendar in the laptop, the desktop PC, your mobile phone or your personal digital assistant (PDA). Moving from one device to another and understanding what the device can do for you is a new experience for the user. Providing the user with a unified user interface has become a new challenge for user interface designers.

In the research project *Always-On,* a service is developed, called Device Unified Services (DUS), which unifies current user's tools to one virtual tool, which is accessible through many devices. This enables users to begin solving a task on one device and continue on another in a different environment. An example is when you talk on a mobile phone on the way to your office, but once you reach the office you would like to switch to the desk phone since it gives you more functionality and better quality. Another scenario is when you browse the Internet on your mobile phone on the train but want to transfer the session to your desk computer at work.

The Always-On project has several partners, companies and universities, in 6 countries in Europe. The project is supported by Eurescom, which is the European Institute for Research and Strategic Studies in Telecommunications GmbH. Founded in 1991, Eurescom provides comprehensive collaborative research management services to network operators, service providers, manufacturers, suppliers and vendors who wish to collaborate on the issues facing the telecommunications industry today.

The complexity of the Device Unified Services raises many interesting issues regarding the user interface design. The same user is using different devices for solving the same task: How consistent does the user interfaces have to be to be usable? Does it have to look exactly the same way on all devices? Does the environment have great influence on how easy it is to solve a task on a particular device? How does the input and output constraints of each device affect the user interface design? How easy is it to transfer a task from one device to another?

The Always-On project took a user-centered approach for modeling the users, their environments, their tasks and the design of the user interfaces for the different devices. Since the tasks require the user to go between devices, we used a navigational model and a more abstract architectural model to model this transfer between devices. In this paper we first describe the user model, the use case model and the user environments. Next we take the use case *Adding a calendar event* as an example to show how the abstract prototypes in the content model, the navigation map and the architectural model is contingent on the devices used. Finally, we discuss how to prepare user tests that include doing the same task on many devices.

## The user, task and environment analysis

This section describes three models that we used to model the users, their tasks and the environment constraints. First we describe the user role model, then the use case model and finally the environment model.

### *User role model*

The typical user is a businessperson who has many appointments to take care of and who is working in a very dynamic environment. The other user roles are his or her spouse and the kids.

**The businessperson role:** This person is running from one meeting to another at different locations and has to be in touch with many people. She wants to be able to work in all these environments including at home and while traveling. She is very technically minded and uses different devices in different environments. She will use DUS many times per day, especially for tasks like adding and viewing a calendar event, making and receiving calls and controlling other family members' use of DUS.

**The spouse role:** The spouse wants to be able to organize the family calendar and check if there are any conflicts to other family members' calendars. The spouse will use DUS many times per week.

**The kid role:** The kids will use DUS both from home and outside home, e.g. from school or friends. They will use it many times each day.

### *Use Case model*

There are 8 main tasks in DUS as follows:

1. Open a Web session
2. Make a call
3. Receive a call
4. Transfer a call
5. Change an environment
6. Add an address book entry
7. Use an address book entry
8. Adding a calendar event

All the user roles will be doing these use cases but with different frequency.

We will take the use case *Adding a calendar event* as an example in the rest of the paper. The Use case description is as follows:

| **Use Case:** | **Adding a calendar event** |
|---|---|
| *User roles:* | All |
| *Purpose:* | To make a calendar event in his personal calendar |
| **User** | **DUS** |
| 1. The user asks for his calendar | |
| | 2. The system displays the Calendar |
| 3. The user selects to add event | |
| | 4. Asks for event information |
| 5. The user fills out the event information | |
| 6. The user ask for contact list | |
| | 7. Displays the contact list. |
| 8. The user selects contacts from the contact list | |
| 9. The user selects modality of the alarm and saves the event | |
| | 10. The system updates the calendar and gives the user a confirmation |

## *Environment Model*

Designers need to fit systems not only to the different tasks and the different user roles but also to the context of use. Similar capability deployed in varied settings may require different solutions to user interface design. Operational factors that can affect user interface architecture and detail design include the following according to [1]:

?? Characteristics of users and user roles
?? Aspects of the physical work environment
?? Features and limitations of operating equipment and interface devices
?? General and specific operational risk factors

.

Referring to the above list, we have described the characteristics of users by the user role model. In the Always-On project it is especially important to describe the next two points that is aspects of the physical work environment and the features and limitations of the operating equipment. The reason is that users are moving from one environment to another and use many devices. Because DUS is not a safety critical application, there are no special operational risk factors.
A user can at any time be in one of five different environments: at his office, at a meeting, at school, traveling and at home. Table 1 contains a short description of the environments and which user roles will be in these environments.

*Table 1: The users and their environments*

| Environment | Description | Buisness-person | Spouse | Kids |
|---|---|---|---|---|
| *Office* | User normally works alone with little distraction. Using his equipment, having a desk in front of him, not noisy, good lighting. | X | | |
| *Meeting* | People sitting at a desk, discussion. Do not want any distractions. | X | | X |
| *School* | Students probably have access to computer rooms, which are similar to the office environment, but maybe noisier and working more in collaboration. | | | X |
| *Travel* | The users will be travelling by train, plane or driving in a car. The environment can be noisy and hard to use equipment, e. g. no table in front of the user. | X | X | X |
| *Home* | Can be quite noisy and distracting, but also very much like the office environment, but more relaxed. | X | X | X |

The users will use five different devices to solve the tasks: a desk phone, a mobile phone, a desktop PC, a laptop and a PDA (personal digital assistant) depending on the environment they are working as table 2 illustrates.

*Table 2: The use of the equipment in each environment.*

| | Desktop PC | Laptop | PDA | Mobile phone | Desk Phone |
|---|---|---|---|---|---|
| *Office* | X | X | X | X | X |
| *Meeting* | | X | X | X | |
| *School* | X | X | X | X | |
| *Travel* | | X | X | X | |
| *Home* | X | X | X | X | X |

Additionally users cannot use all the equipment to solve all the tasks as table 3 shows.

*Table 3: The possible use of equipment to solve the users tasks.*

| Task nr. | | Desktop PC | Laptop | PDA | Mobile phone | Desk phone |
|---|---|---|---|---|---|---|
| *1* | Open a Web session | X | X | X | | |
| *2* | Make a call | X | X | X | X | X |
| *3* | Receive a call | X | X | X | X | X |
| *4* | Transfer a call | X | X | X | | |
| *5* | Change an environment | X | X | X | X | |
| *6* | Add an address book entry | X | X | X | | |
| *7* | Use an address book entry | X | X | X | | |
| *8* | Make a calendar entry | X | X | X | | |

Users can solve all the tasks using desktop PC or a PDA, but users can only make a call and receive a call with a phone; either desk or mobile. Users can perform the task change an environment with a mobile phone. To summarize the PDA and laptop are central equipment, since users can solve all the tasks using PDA or laptop and the users can use the devices in all their environments.

## The user interface design models

First, we describe an abstract prototype with a content model for one use case, namely: Adding a calendar entry. Then the navigation map for two devices is shown, PDA and a laptop. Finally we describe an architectural model for this part of the user interface.

### *Content Model*

A Content model consists of materials and tools that users can apply them. The materials are objects with attributes and the tools are actions. The content model consists of presentations of things that the information system models. The basis of the content model is the conceptual model, i.e. the ideas and concepts of the system. Class diagrams and essential use cases can express the conceptual model. In the next few paragraphs we discuss whether the content model is the same for all devices or if it can be different. We focus our discussion on the materials and the tools of the content model. Later in the section we discuss how we divide the materials into information spaces.

First, we can assume that the content model is the same for all devices given that they support the same tasks. If we support the same use cases with two different devices, the data and the functions are probably the same, so we have the same objects. If we support the action to add an event to a day in a calendar, we need materials for the day and tools to present the addition of an event. All devices must therefore support these materials and tools. Figure 1 shows the content model for the Adding a calendar event task.

Second, we can omit a concept from the content model of a particular device, if it has very much data or many operations. Taking the calendar example, we may not support a year object in a PDA, but only a month or a week.

Third, we can include the same objects but the presentations will be different. In the example of a year, it may simply be an icon or a placeholder without the months in the PDA. As we start to think of the presentations, that is the look and feel, we move away from the content model and start to design the user interface and develop the content model further.

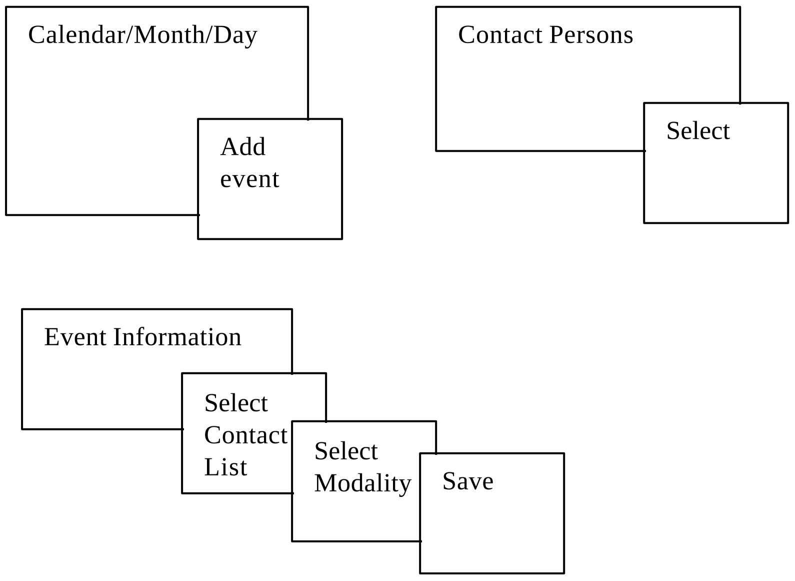

**Figure 1 Content model**

An information space may be customized to an environment. An event that you create for the home may not need as many operations as the one you need for the office. For example if you were to book a picnic with your family you may not need security clearance or permission from your boss. This may however be the case when you book a meeting on confidential subjects within the company.

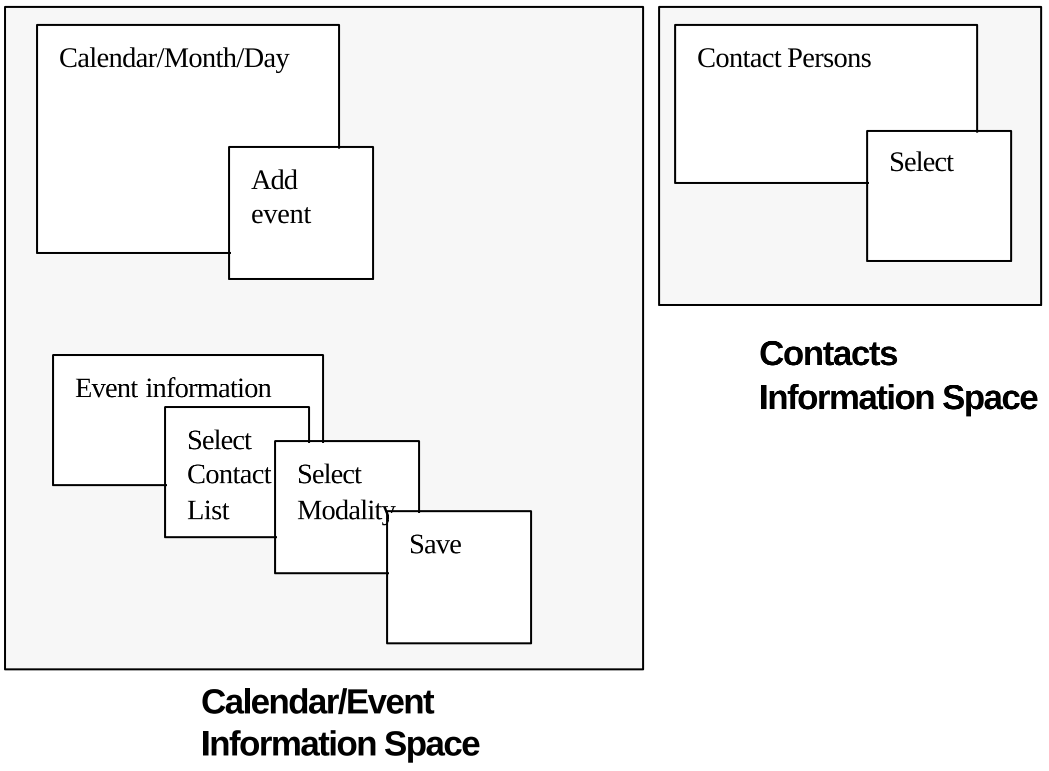

**Figure 2: Two information spaces for a desktop PC**

Figure 2 shows the contexts or information spaces of the user interface objects for a desktop PC. Each rectangle represents a context. Figure 3 shows the information spaces for a PDA. We need to break the content model into more information spaces in a device that has less screen space.

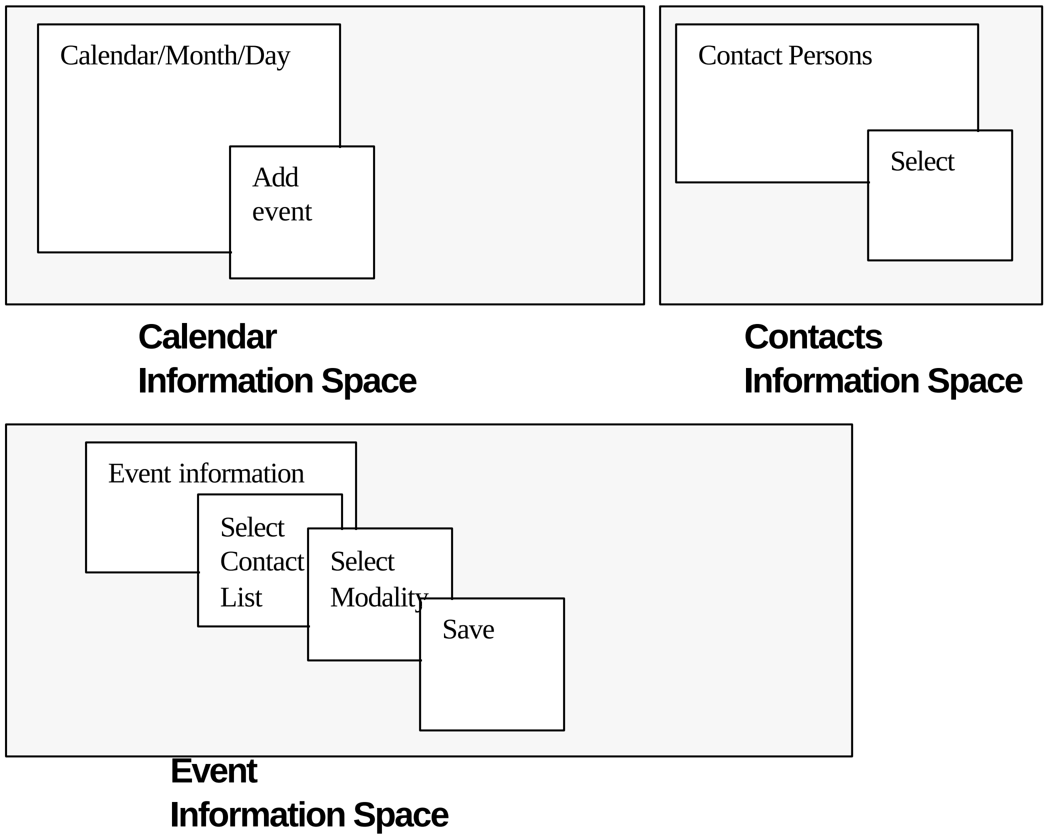

**Figure 3 Information Spaces for PDA**

## *Navigational models*

In ordinary context navigation maps, information spaces become concrete screens, dialogue boxes or windows. We navigate between information spaces or these concrete user interface elements. Figure 4 shows an example of a navigational map for adding an event to a calendar in a PDA.

A navigational model is a kind of a state diagram, where the information spaces represent states. It is very useful to draw a navigational map during design or reengineer it from a user interface. A navigational map emphasizes showing tools that result in transitions between information spaces but not transitions within an information space. A more detailed state diagram can show different states of a single context. This is for example useful to model if we perform an action on an object that affects the allowable operations on that object. An example is when we mark a calendar day to be a holiday we cannot create any work-related events on that day.

We can split one concept into two user interface objects in two different information spaces. A week may be split into two information spaces if space doesn't permit us to have it all on one screen.

In an application where we have many devices, contexts can be distributed over several devices. A user then can navigate between devices. The navigation between devices occurs normally when you exit one environment and enter another. The reason you change devices is when a preferred device becomes available. An example of this may be when you are registering an event on a PDA in your car and then go to your office and want to add more members from the Address book on your desk PC.

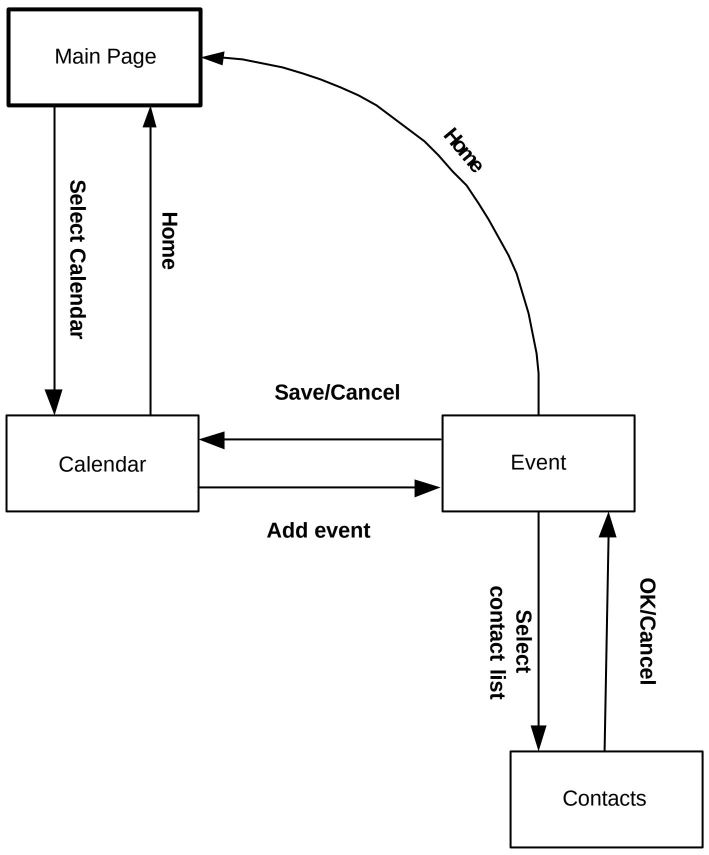

**Figure 4 Navigational map for a PDA**

## *Architectural model*

Architecture of a user interface consists of the presentations of information, or the user interface elements, their structure, the relationship between them and their mapping to the physical devices. In software architecture, we decide the breakdown of the functionality into components and we decide e.g. on what hardware the database is stored and on which hardware the application server is stored.

The user interface architecture describes relationships between information spaces and what devices are responsible for the different information spaces. If we have many information spaces we can choose to cluster them into more abstract user interface components to get a better overview of the complex user interface. When we have only one device as we have had for a long time, the hardware mapping is trivial. When we have multiple devices the architecture plays a more important role.

An architecture can be described with a set of all navigation maps. An architecture gives an overview of the user interface. Therefore, it is better if we abstract some navigational contexts into larger components. Figure 5 illustrates an architecture for the PDA and Desktop PC information spaces. It shows how the user can transfer from one device to the other and go to the corresponding information space. The information space on the destination device may not have all the materials and the tools of the information space as the source information space. The reason is that the materials may be devided differently into information spaces for different devices.

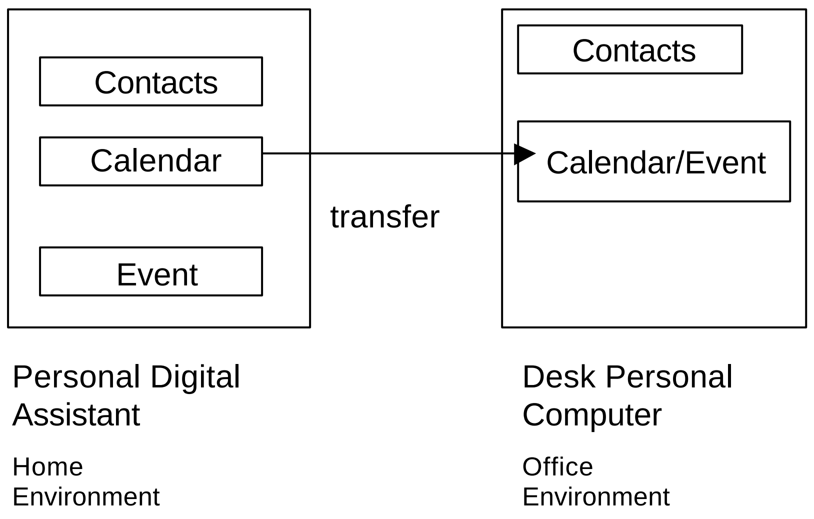

**Figure 5 Architecture**

## Usability testing

The DUS has very limited functionality. The users are only able to perform 8 main tasks, but there are three user roles, five environments and five different devices. This leaves us with a lot of possible ways of testing the usability. The goal for the usability testing is that all tasks are tested at least with 5 users and the same user does at least four tasks twice with two different devices. So we can have some good material on what device was easier to use to solve a particular task.

Obviously the tasks will be described in a very similar way for different devices.  One example can be: You are at a meeting and you want to schedule a meeting next Monday at 10:00 am by using your PDA instead of your laptop. The kids would not like this description, and as seen in the environment model they are never in the meeting environment. So the task description has to map the user role domain knowledge.

Testing the same task but in a different environment can be hard to simulate in a laboratory setting. Simulating the office or the school environments will be easy, but the home environment and meeting environments are a little more complex and especially the travelling environment will be challenging. So it is still a question whether testing a task with a single device but in two different environments will be included or not.

## Conclusion

After using the models we find the user model, the task model, the content model and the navigational model very helpful. Doing one content model for each device is a very good way of structuring materials and tools in an abstract way and doing one navigation model for each device is also very helpful. The architectural analysis gives a better overview in a complex environment. More work is needed on formal notation on user interface architecture.

In a very flexible system like DUS, analysing the environments is fundamental and further work is needed to structure that analysis. The operational model in [1] does not cover the complexity of describing many environments for the user roles nor how to express what user roles are included in each environment. Additionally pointing out for each environment with what devices it is possible for the user roles to solve the different tasks is crucial.

Finally there are many possible ways of testing the usability of a tool, which is used by many different user roles, in different environments with various devices. This is so even if the tasks that the user is able to perform are very few. Some of the environments will be hard to simulate in the laboratory. We need to carefully describe the user tasks in the usability test so that they fit the user domain knowledge.

## References

[1] L. L. Constantine and L. A. D. Lockwood. *Software for Use: A Practical Guide to the Models and Methods of Usage-Centered Design.* Reading, MA: Addison-Wesley, 1999.