Towards a Notation for the Modeling of User Activities and Interactions Within Intelligent Environments

Christoph Stahl

Saarland University, Stuhlsatzenhausweg 36.1
66123 Saarbrücken, Germany
stahl@cs.uni-sb.de

The design of ubiquitous computing systems for pedestrian assistance in large and complex environments, such as airports, museums or conferences, poses new challenges to existing development processes. The designer of such a system has to specify typical activities within the environment and how the system supports them, and decide which sensors and devices will be used for interaction. We propose a new design process, which extends the scenario-based design approach through the use of a more structured model of user activities and human-computer interaction in intelligent environments.

Keywords: Software design process, activity model, intelligent environments

INTRODUCTION

Consider yourself in the position of a software developer, who has been hired to enhance the environment of a large airport with an ubiquitous computing infrastructure in order to assist pedestrians in situ by their typical activities, such as navigating through the check-in procedure or spending leisure time with shopping. How would you proceed with the design of such a system, if all choices are left to your expertise? You were confronted with a broad design space, spanning from small personal devices like mobile phones to large, situated public displays and multiple modalities for interaction, and various sensor technologies. Since few requirements can be initially specified, you would probably follow a user-centered design approach, based on scenarios, prototypes and user studies. Most likely, you would split up in separate teams, who come up with appropriate assistance applications for each activity, such as navigation and shopping. However, this approach bears the risk of lacking integration between the resulting subsystems, causing problems for the user, who pursues multiple activities at the same time and thus has to change focus between multiple user interfaces and devices. Therefore we believe the system should be modeled and designed as a whole, including all activities and interaction artefacts.

RELATED WORK

We will now briefly summarize existing software development processes and user activity models.

Software Development Processes

Traditional Software Engineering tries to overcome the difficulties in large-scale software development through the Waterfall model: An initial specification of requirements guides the development process through the design and implementation phases and allows to test and validate the final product against them. The pitfall of this method is that for innovative projects no specific requirements can be formulated, and the product might not be accepted by the user. Hence Norman and Draper introduced the term User-Centered Design [6], where the users' activities and goals are analyzed first in order to specify an application that will fit the users' needs. Following this idea, the Scenario-Based Design [2] methodology proposes to work out story-like scenarios in cooperation with the user, which describe how a potential system would assist them in their respective situation. The *RESCUE* (Requirements Engineering with SCenarios in a User-Centered Environment) [5] process combines Scenario-Based Design with concepts from Task Analysis for Requirements Analysis in the domain of Air Traffic Management. The goal is the validation of use-cases and a more complete specification of use-case models. It generates human activity descriptions, based on pre-defined templates, which describes goals, actions, ressources, contextual features and constraints. Actions are furthermore distinguished between physical, cognitive and communicative acts.

User Activity Models

In [6], Norman formulates a theory of action, which approximates user activities by seven stages: establishing the goal, forming the intention, specifying the action sequence, executing the action, perceiving the system state, interpreting the state, evaluating the system state with respect to the goals and intentions. Crowley et al. [3] propose an ontology for the modelling of the users' context and situation, where states are defined by observable properties of the world, and a universe is a graph in which states are connected through actions; to attain a goal state, the user must perform a sequence of actions; the association between the current state and the goal state is a task. The current set of tasks is the user's activity. The concept of Activity Zones [7] is focused on location and defines regions in which similar daily human activities occur.

TOWARDS A UBICOMP DEVELOPMENT PROCESS

The typical development process in the ubicomp research community is triggered by technological innovation and goes bottom-up: As new artifacts and sensors become available, an interesting application domain is chosen and a prototype is implemented and evaluated. We argue that the design of real-world applications will rather follow a top-down approach, where a given environment is analyzed for typical users' activities, before in the next stage the broad design space of technological choices is searched for an appropriate solution. Baber describes in [1] such a process for wearable systems, where in the first phase the domain specific requiremens are generated from a scenario, that describes activities and their context-of-use together with some choices for interaction modalities. In the next phase, a so-called modality matrix is used to rate all possible modality choices against the initial requirements in order to reduce the design options.

Besides the modalities, in ubiquitous computing scenarios are even more design decisions about computing hardware and user interface technology to be made: the options comprise mobile, wearable and stationary artefacts, embedded (invisible) processors and a variety of additional sensors (vision, touch, speech) to support location-awareness and multimodality. Therefore we propose the following design process, which is similar to Baber's, but requires a more elaborated and formal model:

- Describe the requirements through scenarios
- Transform the scenarios into a more formal model of activities and actions.
- Decide for human-machine interaction technology.
- Transform the formal model into a system design.
- Build and evaluate a working prototype using toolkits.
- Iterate the process, or implement the final system.

What Should be Represented by the Model

The model which is to be described in the second phase of our proposed ubiquitous computing design process should unify the ideas of Norman's activity theory, Crowley's contextual model and Koile's activity zones (see related work section). Since there are too many details to argue about, we will not try to give a specific schema for such a model in our paper. Instead, we suggest a set of questions, which will guide the designer of an ubicomp system to create her own model that represents all necessary facets:

What are the user's **goals, activities and actions** in the scenario? We suggest to base the distinction between an activity and an action on their temporal granularity: an activity takes a time span, in which actions occur instantaneously. On a finer level of abstraction, actions itself can be treated as sub-activities, which can be recursively described through actions.

What are the **preconditions and effects** of an action? This question helps to identify problems and opportunities for user assistance regarding the

- Location of the user
- Time constraints
- Resources (physical objects or information)
- State of the environment (situational context)?
- State of the assistance system (transitions, operations)?

Which actions should be mapped to **interactions** with the assistance system? We can distinguish between

- Incidental interaction *"actions performed for some other purpose or unconscious signs are interpreted in order to influence/improve/facilitate the actors' future interaction or day-to-day life"* (A. Dix [4])
- Implicit interaction *"Implicit human computer interaction is an action, performed by the user that is not primarily aimed to interact with a computerized system but which such a system understands as input"* (A. Schmidt [9])
- Explicit interaction

Who **initiates** the interaction, the user or the system, and what is the **intention**?

- Typical intentions for user-initiated interactions might be to
  - satisfy an information demand
  - create a note (mental support through externalization)
  - create a reminder/alarm (and suspend the current activity)
  - communicate or exchange documents between users
- Typical intentions for system-initiated interactions might be to provide a
  - instruction step for task completion (in a procedural activity)
  - notification message and/or alarm (to begin or resume an activity)

What is the **flow of information** for an interaction? Multiple information sources and destinations are allowed, and they can be either

- Virtual (databases, documents or media streams) or
- Physical interaction devices, e.g. displays or buttons

What **types of actions** are performed by the user and the system during an interaction?

- Physical actions (e.g. move, grab object, interact with artefact)
- Communicative actions (talk)
- Cognitive actions (remember, recognise, recall, plan, decide, perform, check. validate, mentally acknowledge, read, write)
- Computational actions (database lookup, presentation generation, planning)

Example Scenario: Designing for User Assistance in an Airport Environment

We will now give an example on how the proposed methodology could be applied to design a user assistance system in an airport environment. In [10], we introduced a scenario where a user is going to buy a digital camera in a duty free shop during his stay at an airport. We have also described an intelligent environment and infrastructure, which proactively assists its users in their shopping and navigation tasks. Whereas each of the implemented assistants quite successfully support a single activity, they have not been integrated properly yet. In order to redesign the system, we will apply the proposed methodology and questions from the previous section.

Step 1: Describe the requirements through scenarios

We begin the design process with two scenarios, which are introduced in Box1, and describe typical problems that travelers have in certain situations. They represent the requirements and opportunities for user assistance within an intelligent environment.

The Passenger Process

Tim is <u>traveling</u> from Tokyo to Frankfurt, and has just arrived at the airport. He <u>grabs a trolley</u>, and moves on towards the main entrance hall. There he checks the timetable for his flight number and destination, and <u>moves on to the terminal building</u>. Unfortunately, the <u>check-in desk</u> is not in the same place where it used to be last year, and he has to <u>locate</u> it quickly before it closes. In the hurry, he almost can't find the printout of his <u>online ticket</u>, and would probably forget the <u>gate number</u>, if the clerk does not write it down on his <u>boarding pass</u>. But the excitement turns into boredom, as his flight is delayed for 2 hours due to technical issues with the aircraft. After a <u>quick meal</u> in his favorite restaurant chain, Tim decides to pass through the <u>security check</u> and proceed directly to the duty free <u>shopping</u> zone in order to see whether he can save some money there. Concerned with a <u>buying decision</u>, time passes quickly and as he checks the timetable, he recognizes that the <u>gate</u> has changed to the opposite side of the airport. Ten minutes later, he is almost the last passenger <u>boarding</u> the plane.

Duty-Free Shopping

Tim has one hour to spend in the duty-free shopping area. He has been interested in digital SLR cameras for some time, and is now looking for a <u>bargain</u>. However, the model he had in mind seems outdated in <u>comparison</u> to new models. He <u>picks up</u> two boxes, but they don't show much technical details. Tim misses the Web, where he usually obtains all the information he requires for a <u>buying decision</u>, ranging from interactive presentations to price comparisons and buyer comments.

Box 1: The passenger process and duty-free shopping scenarios

Step 2: Transform the scenarios into a model of activities and actions

We will now identify the user's activities and actions from the two scenarios, as shown in Box 2. We describe the preconditions and effects of an action, which will reveal possible problems and opportunities for assistance. For example, the check-in action requires knowledge of the ticket code and the user will receive a boarding pass. Since most actions have a certain location as precondition, we separate them as wayfinding activities. Now we have to choose actions, which are candidates for interactions between the user and the assistance system. We have marked them with (S) for system initiated interactions such as notifications, (I) for implicit and (E) for explicit user interaction with the system.

The Passenger Process

- Get a trolley
- (S) <u>check-in</u> and register: at least 40 min before takeoff, requires passport and ticket or code, user receives boarding pass with information about gate and time
- (E) Quick meal / <u>consumption</u>: requires survey of choices
- Security check: requires passport and boarding pass, passengers have to queue in line and wait for some minutes
- Duty-free shopping: is an activity itself
- (S) <u>Boarding</u>: hard deadline, requires boarding pass, achieves goal

Duty-Free Shopping

- (I) Upon <u>entering</u> a shop, check for interesting sales and new products
- (I) <u>Pick up</u> a product
- (I) <u>Compare</u> products for informed buying decision
- Convert currencies

Wayfinding

- (S) <u>Locate</u> one of the following destinations:
  - terminal building
  - check-in desk
  - security check
  - shops
  - gate
- (E) <u>Browse</u> places for consumption (restaurants, shops)
- (I) <u>Roam</u> around the place

Box 2: Summarizing the activities and actions from the scenarios

Step 3: Decide for human-machine interaction technology

Now that we have identified opportunities for interaction, we have to decide which device classes and technologies are applicable for each action. First, we briefly discuss the pros and cons of three device classes: personal (PDA and mobile phone), public (information kiosks), and ubiquitous computing artifacts, where we consider an instrumented trolley. In the next step we compare their sensing capabilities to support implicit interaction. We need a means to sense the user's position in order to recognize the actions of roaming and entering a shop. In the shopping scenario, we have to identify the products picked up and put back by scanning their barcode, or more convenient by reading RFID tags. Table 1 shows the availability of typical technologies for the different device classes, and we decide for the instrumented trolley.

Personal Devices

PDA

+ no investment for the airport required, good privacy
- few people have one, requires software download, stylus difficult to use on the move or with baggage, connectivity and configuration issues

Mobile phone

+ easy to use with one hand, widespread
- not every phone is programmable, difficult to localize, data transfer is expensive

Public Devices

Public Displays / Information Kiosks

+ large screen space, touchscreen easy to use, already many displays available
- restricted availability, difficult to design for multiple users, difficult to notify user

Ubiquitous Computing Artifacts

Instrumented Trolley with graphical display and RFID reader

+ User can interact with the device on the move, convenient to use
- expensive investment, requires charging mechanism, may be damaged

Sensing		PDA	Mobile	Kiosk	Trolley
Positioning	WLAN	+	-	n. a.	+
	RFID	-	-	n. a.	+
	Cell	-	+	n. a.	-
Gesture	Barcode	+ camera	+ camera	+ scanner	+ scanner
	RFID	-	-	+ reader	+ reader

Table 1: Availability of sensing technologies for the different device classes

We can now transform the identified interactions into more formal, template-based models for each activity. Table 2 describes the actions related to the passenger process, and defines their preconditions and effects on both the world state and the system state. The table further decomposes each interaction into physical and cognitive actions of the user (shown on the left) and computational actions of the system (shown on the right). The model allows us to consider how the instrumented trolley could support each step of the interaction. The action to get a trolley is user-initiated, and the user explicitly interacts with the trolley to log on to the system by scanning his customer card. Then the system takes the initiative and proactively instructs the user step-by-step through the process. Each action begins with a notification message, possibly combined with an alarm signal, and the presentation of required information such as the ticket code or gate number.

Activity: Passenger Process		**User**	**System**
Goals and Intentions		Get on the plane	Guide the traveller step-by-step through the procedure
Action ”get a trolley”	Precondition	User is at the entrance	State s0
	Phys. Act	Pickup trolley, log on	Scan customer card
	Comp. act		Retrieve flight information from airport database
	Phys. act		Display welcome message
	Cog. act	-	
	Effect	Ready to check-in	State s1
Action ”check-in”	Precondition	Ready to check-in	State s1
	Comp. act		Set navigation destination to “check-in counter”
	Phys. Act.		Remind the user of his ticket-code and passport; Display remaining time
	Cog. Act.	Follow instructions	
	Effect	Check-in completed	State s2
Action ”security check”	Precondition	Check-in completed	State s2
	Phys. Act	-	
	Comp. Act		Estimate remaining time; Set navigation destination to “security check”
	Phys. act		Remind the user of passport Display remaining time
	Cog. Act	Follow instructions	
	Effect	Ready for shopping	State s3
Action ”boarding”	Precondition	Check-in completed	State s3
	Phys. Act	-	
	Comp. Act		Estimate remaining time; Set navigation destination to “gate”
	Phys. act		Remind the user of boarding pass and gate number
	Cog. Act	Follow instructions	
	Effect	Goal achieved	State s0

Table 2: A template-based interaction model for the passenger process assistant.

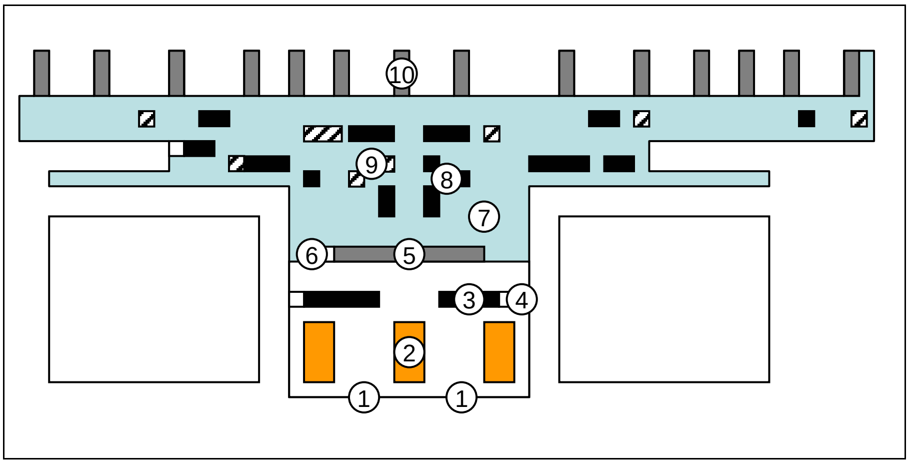

Figure 1: Spatial map of the airport, indicating the locations of the passenger process.

The activities of the passenger process have a very strong dependency on location, and often involve extensive walking distances. We suggest supplementing the interaction model with a spatial model of the environment. In Figure 1, a simplified map is shown that indicates the locations of each action in the passenger process. We can spatially reference the actions to the map:

1. entrance, pick up a trolley
2. check-in counter area
3. consumption area, e.g. having a quick meal
4. bank, ATM and currency conversion
5. security check
6. customs and taxes
7. waiting area
8. duty-free shops
9. restaurants
10. gates, boarding

We suggest using the *Yamamoto*[1] map modeling toolkit for the graphical annotation of digitized ground plans, which has been developed in our group. It allows marking up the outlines of places through polygons and annotating them through symbolic names. So the places can be grouped into activity zones, which can be used to implicitly trigger application actions and to present information to the user, as described by Koile in [7]. Whereas it takes some additional effort to create such a spatial model early in the design phase, it provides for an exact specification of location-based system behavior and it will pay off later during the implementation of a navigational assistance subsystem.

[1] Yamamoto Map Modeling Toolkit, Website: http://w5.cs.uni-sb.de/~stahl/yamamoto/

Since many users have difficulties in reading graphical maps and the meaning of existing signage is not always clear, the assistance system should provide individual navigational aid to the user. In Table 3, we describe in detail how the instrumented trolley supports the user in his wayfinding tasks. The first action "system sets destination" is activated through the passenger process actions, such as "security check", in order to proactively navigate the user through the process. The second action allows the user to browse through interesting destinations, e.g. quick-serve restaurants or gift shops. It is also useful in the duty-free shopping area. The third action is implicitly triggered through motion as the user reaches a navigational subgoal. The position of the trolley might be determined by the trolley itself reading RFID tags placed in the environment, or vice versa by the environment tracking an RFID tag inside the trolley[2]. WLAN positioning techniques such as *Ekahau*[3] would be an alternative solution. New instructions are presented to the user through the trolley's display and possibly audio, until the user has reached the destination.

Activity: Wayfinding		**User**	**System "Pers. Navigator"**
Goals and Intentions		Find a certain destination	Provide directional aid
States and Conditions			n0: no destination set n1..sn-1: user at subgoal nn: user at destination
Action "system sets destination" (check-in desk, security check, gate)	Precondition		n0
	Comp. action		Initialize selected route
	Phys. action		Generate first instruction step
	cog. / sense	Read/Listen to instruction	
	Effect	User moves towards first subgoal	n1
Action "browse Destinations"	Precondition	Check-in completed	n0
	phys. action	Browse destinations	
	Comp. action		Initialize route
	Phys. action		Generate first instruction step
	cog. / sense	Read/Listen to instruction	
	Effect	User moves towards first subgoal	n1
Action "subgoal"	Precondition	User reaches subgoal i	state ni
	Phys. Action	Moving the trolley	Receive RFID beacons
	Comp. Action		Generate next instruction step
	Phys. Action		Present instruction for next subgoal; speech synthesis
	Cog. Action	Read/Listen to instruction	
	Effect	goal reached; or User moves towards next subgoal	n0 if destination is reached; ni+1 else

Table 3: A template-based interaction model for the wayfinding activity.

[2] appropriate tracking technology is offered by UbiSense, website: http://www.ubisense.net

[3] website: http://www.ekahau.com

Activity: Camera Shopping		**User**	**System “Shop Assist”**
Goals and Intentions		Buy digital camera: balance low price against requirements	Provide product specification
Action ”enter shop”	Precondition	User is outside	State p0
	Phys. Act	User enters the shop	
	Comp. Act		Lookup user profile; Lookup product database
	Phys. act		Present sales offers
	Cog. Act	Browse through offers	
	Effect	User is ready to shop	State p1
Action ”pickup product”	Precondition	User is ready to shop	State p1
	Phys. Act	Pickup first product from shelf	Recognize product barcode
	Comp. Act		Lookup product database
	Phys. act		Present product information
	Cog. Act	Read product information	
	Effect	User knows camera details (for buying decision); User holds one product	State p2
Action ”request details”	Precondition	User has located a camera on the shelf, empty hand	State p2
	Phys. Act	User Asks: “How many Megapixels does this camera have?”	Speech recognition
	Comp. Act		Lookup camera database
	Phys. act		Present requested details
	Cog. Act	Read product information	
	Effect	User knows camera details	State p2
Action ”pickup alternate product”	Precondition	User holds one product	State p2
	Phys. Act	Pickup second product	Recognize product barcode
	Comp. Act		Lookup product database
	Phys. act		Present comparison information
	Cog. Act	User makes decision	
	Effect	Ready to buy the product	State p3

Table 4: The interaction model for the shopping assistant.

Table 4 describes the necessary interaction steps for the envisioned user assistance in the shopping scenario. It is similar to our *Smart Shopping Assistant*, which has been presented in [11] and uses a shopping cart that has been equipped with a RFID reader to recognize the products inside. We make the assumption here that RFID tags will replace the traditional barcode in the future. Today we would have to use a barcode scanner instead. The *ShopAssist* [8] application demonstrates how speech and gesture combined can be used for rich multimodal interaction.

System Design, Implementation and Evaluation

The interaction model that we have developed so far can be refined and enriched with more details, until it represents a complete system specification, from which a user interface can be designed and a first prototype can be implemented for evaluation.

DISCUSSION

We have proposed an extension to the scenario-based design methodology for the design of ubiquitous and pervasive computing applications, which is based on a more formal model of the interaction between the user and an intelligent environment. Based on scenario descriptions, we identify activities and actions and map them into interactions between the user and the assistance system to be designed. Another step is the careful selection of devices and sensing technologies. Instead of a fixed schema for an interaction model, we have suggested a set of questions, which will guide the designer of a system towards his own model. We have demonstrated how such a model might look like by the example of an airport assistance service for travelers, where an instrumented trolley guides the passenger through the check-in process.

REFERENCES

[1] Baber, C. et al. Contrasting Paradigms for the Development of Wearable Computers. IBM Systems Journal, 38(4):551-565, 1999.

[2] Carroll, J. Making Use: Scenario-based design of human-computer interactions. MIT Press, 2000.

[3] Crowley, J., Coutaz, J., Rey, G. And Reignier, P. Perceptual Components for Context Aware Computing. Ubicomp, Springer, 2002.

[4] Dix, A., Beyond Intention - pushing boundaries with incidental interaction. Proceedings of Building Bridges: Interdisciplinary Context-Sensitive Computing, Glasgow University, 9 Sept 2002

[5] Maiden, N. and Jones, S. Dependability in RESCUE: A Concurrent Engineering Approach to the Specification of Requirements for Air Traffic Management.

[6] Norman, D. and Draper, S. User Centered System Design, Lawrence Erlbaum Associates, Inc., 1986.

[7] Koile, K., Tollmar, K., Demirdjian, D., Shrobe, H. and Darrell, T. Activity Zones for Context-Aware Computing. In Proceedings of Ubiquitous Computing, 5th International Conference, Seattle, Springer, pages 90-106, 2003.

[8] R. Wasinger, M. Schneider, J. Baus, and A. Krüger. Multimodal Interaction with an Instrumented Shelf. In Artificial Intelligence in Mobile Systems 2004 (AIMS 2004), number ISSN 0944-7822, pages 36-43. 2003.

[9] Schmidt, A. Implicit Human Computer Interaction Through Context. Personal Technologies, 4(2&3), Springer-Verlag, pp. 191-199, June 2000.

[10] Stahl, C., Baus, J., Brandherm, B., Schmitz, M. and Schwartz, T. Navigational- and Shopping Assistance on the Basis of User Interactions in Intelligent Environments. Proceedings of the IEE International Workshop on Intelligent Environments (IE 2005), University of Essex, Colchester, UK, 2005.

[11] Stahl, C., Baus, J., Krüger, A., Heckmann, D., Wasinger, R., and Schneider, M. REAL: Situated Dialogues in Instrumented Environments. Workshop on Invisible and Transparent Interfaces at AVI 2004, Gallipoli, Italy, 2004, pp. 10-15.