A reference model for situation-aware assistance

Thomas Kirste, Fraunhofer IGD Rostock
kirste@rostock.igd.fhg.de

Abstract As computers are becoming more and more ubiquitous, moving from the desktop into the infrastructure of our everyday life, they begin to influence the way we interact with this environment – the (physical) entities that we operate upon in order to achieve our daily goals. The most important aspect of future human-computer interaction therefore is the way, computers support us in efficiently managing our personal environment.

Conventionally, human-computer interaction looks at a process, where only two partners are involved: the human and the computer. However, looking at the computer as a *mediator* between the user and his environment, we have to acknowledge a more complex communication process.

This paper proposes a reference model that identifies the fundamental components which are involved in human-computer-environment interaction.

1 Motivation

A human being's daily activities – professional or private – are based on a broad range of interactions with numerous external objects: discussing project plans with colleagues, setting up a multimedia presentation in the conference room, editing documents, delegating travel planning to a secretary, driving a car, buying a ticket from a vending machine, visiting an exhibition, controlling the TV at home, etc.

As computers are becoming more and more ubiquitous, moving from the desktop into the infrastructure of our everyday life, they begin to influence the way we interact with this environment – the (physical) entities that we operate upon in order to achieve our daily goals. The most important aspect of future human-computer interaction therefore is the way, computers support us in efficiently managing our personal environment. This might be called the *ecological* level[1] of user-interface design.

At the ecological level, we look at future developments from the perspective of helping a user in achieving his individual goals and purposes by providing computer-based assistance for interacting with his personal environment.

The goal is to have the computer acting as a *mediator* between the user and the environment – e.g., giving the user hints for operating an obstinate ticket vending machine, reminding him of things he wanted to tell a colleague just approaching across the corridor, etc. Because the machine has no *direct* control over the human/environment interaction, it needs to achieve the desired effects indirectly through the other interfaces: i.e., by sensing the environment, by cleverly guessing the user's goals and future interactions with the environment, and by proactively providing the user with information needed for those activities (or by actively controlling the environment itself).

Conventionally, human-computer interaction looks at a process, where only two partners are involved: the human and the computer. However, looking at the computer as a *mediator* between the user and his environment, we have to acknowledge a more complex communication process.

Central challenges for providing such *personal ubiquitous assistance* are:

- Understanding the user's goals and the specific strategies employed by the user for achieving these goals.

[1] "Ecology is the scientific study of interactions of organisms with one another and with the physical and chemical environment." So this notion quite well captures the essence of the above discussion.

- Sensing and understanding the user's personal environment and the ways the current environment influences his activities and strategies.

In addition, in order to minimize the cognitive (and sensomotorical) gap between human/computer interaction on the one side and human/environment interaction on the other side, *natural* (anthropomorphic) *interaction* should be supported: Multimodal interfaces lead the way with features such as:

- speech input (command phrases as well as natural dialog) and output,
- video based interaction (e.g., gesture and position recognition),
- avatars as graphical output metaphors,
- haptic feedback for buttons and knobs (e.g., in a car environment).

These technologies are implemented as a modular system from which different user interfaces depending on the requirements of certain appliances can be build.

Two major application areas currently investigate the use of natural interaction and intelligent assistance for creating ecological interfaces:

- On the professional side, the office environment will be reorganized by means of agent-based intelligent assistant systems in order to reduce time consuming routine tasks and unnecessary interrupts. Progress in the fields of delegation-based interfaces, intelligent assistant systems, man-machine-communication, mobility, and security will lead to new *multimedia workspaces* – such as investigated in the project MAP [8].
- On the other hand, *interactive appliances* will allow a unified and simplified access to the gadgetry of modern life. Having a single, personal control with a customized user interface for interacting with appliances at home (e.g. audio/video-appliances, washing machine), in the streets (e.g. ticket-vending machines), on the road (e.g., car stereo and air conditioning), on-site or remote, will make us feel at home everywhere. This is the focus of the project EMBASSI [4, 3].

As outlined above, the ecological level is concerned with exploiting new ways for helping a user in interacting with the various objects in his personal environment in order to achieve his individual goals and purposes effectively and successfully. Common to both application areas outlined above is the *proactive* and *environment-sensitive* nature of the underlying solution concepts. The systems need to use knowledge on the user's individual goals, his strategies for achieving them, and the dependencies of those strategies on the user's current personal environment for anticipating the next steps of the user. In doing this, those systems effectively become intimate personal assistants to the user – up to the degree that they might be regarded as a kind of mental prosthesis: (simplified) projections of the user's mind and his specific cognitive structures onto a different (electronic) substrate.

The goal of this paper is to propose a reference model that identifies the fundamental components which are involved in human-computer-environment interaction.

In the next section, we will outline the essential notions of this concept.

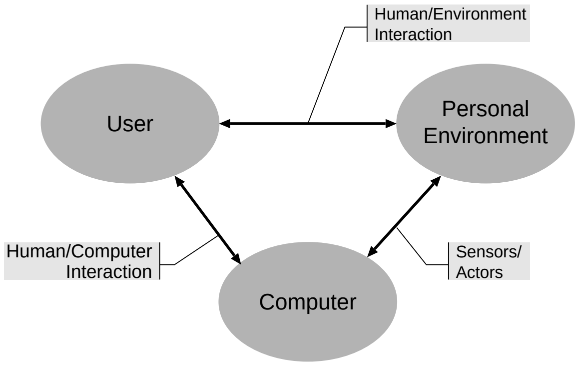
2 Overview of the SAA-model

The SAA-model (SAA stands for **S**ituation **A**ware **A**ssistance) looks at technical systems (specifically computerized assistants) that support the user in *controlling* his personal environment. The SAA-model is based on the following concepts:

- The notion *control* is only meaningful for objects that have a **state**. To **control** an object means to be able to (intentionally) change its state.

  The SAA-model describes systems that support the user in changing the state of his environment.
- For changing the state of the personal environment, there exists a set of **actions** that can be performed within the environment. However, performing actions is not the user's *goal*, but rather a *means* for achieving such a goal.

Figure 1 The ecological interface



- The user's **goal** is to have the environment assume a specific state. A certain sequence of actions may produce this state, but maybe a completely different action sequence will give the same result. (In general, the goal of the user is not given by one specific state, but rather a *set* of permissible states.)
- In order to reach the user's goal, a *plan* has to be developed. A **plan** is a sequence (more general: partial ordered set) of actions, the execution of which will transform the current state into one of the goal states.
- To **execute** a plan means to perform the actions specified by the plan in an order that is compatible with the plan. (If the order is linear, the execution is in sequence.)

The task of an assistance system is to support the user in reaching his goals. This encompasses the *complete* process, beginning at the goals and ending at the execution.

3 Detailed Discussion

The SAA-model is defined using the Z-Notation [9]. The static correctness of the specification has been verified with the `ztc`-Tool [5].

3.1 Supporting definitions

We introduce some supporting definitions in order to simplify the discussion below.

$$[X]$$
$$id : X \to X$$
$$fold : \mathrm{seq}(X \to X) \to (X \to X)$$

$$\forall x : X \bullet id\ x = x$$

$$\forall f : X \to X;\ s : \mathrm{seq}(X \to X) \bullet$$
$$fold\ \langle\rangle = id[X] \land$$
$$fold\ (\langle f \rangle \frown s) = (fold\ s) \circ f$$

id is the identity function; $fold$ takes a sequence of functions and returns a function that represents the sequential composition of all the functions in the list.

In addition, we need a function $frob$ that takes list of functions and a list of arguments, returning the list of results created by applying each function to the corresponding argument. Finally, map simply takes a function and a list of items and returns a list created by applying the function to each item in the argument list.

$$
\begin{array}{l}
[X, Y] \\
\hline
frob : \mathrm{seq}\, X \rightarrow \mathrm{seq}(X \rightarrow Y) \rightarrow \mathrm{seq}\, Y \\
map : (X \rightarrow Y) \rightarrow \mathrm{seq}\, X \rightarrow \mathrm{seq}\, Y \\
\hline
\forall\, xs : \mathrm{seq}\, X;\ fs : \mathrm{seq}(X \rightarrow Y) \bullet \\
\quad frob\ xs\ fs = (\lambda\, i : \mathbb{Z} \bullet (fs\ i)\ (xs\ i)) \\
\forall\, f : X \rightarrow Y;\ s : \mathrm{seq}\, X \bullet \\
\quad map\ f\ s = f \circ s
\end{array}
$$

3.2 The kernel

3.2.1 The world and its actions

The environment – rather: the world according to the machine – is defined by a given set of states, $SystemWorld$.

$$[SystemWorld]$$

At this level of abstraction, we consider $SystemWorld$ as opaque. It not necessary – at this place – to further analyse the inner structure of the environment state (such as dimensionality etc.).

Actions on the environment are simply functions mapping states to states. Action sets ($Actions$) are sets of actions.

$$Action == SystemWorld \twoheadrightarrow\!\!\!\!\!+ \; SystemWorld$$

$$Actions == \mathbb{P}\, Action$$

3.2.2 The user and his goals

User goals – as far as the system is concerned – are described by state sets (the set of permissible states that represent the user's goal).

$$SystemGoal == \mathbb{P}\, SystemWorld$$

At the current level of abstraction, it is not necessary, to investigate suitable means for representing such state sets. (Such means would make use of the potential dimensionality of $SystemWorld$.)

It is impossible to think about assistance system without having a precise model of the user. From the viewpoint of the SAA-model, a user is represented by a function that maps a sequence of environment states to a sequence of goals.

$$User1 == \mathrm{seq}\, SystemWorld \rightarrow \mathrm{seq}\, SystemGoal$$

In a less abstract formulation: the user is an entity that is able to observe the environment as it changes over time. As time passes (and as the environment changes), goals appear in the user's mind that represent the things the user wants to happen in the environment.

Note that $User1$ only represents the will-functions of the real user. Other functions – e.g., reasoning, planning, and physical interaction – are contained in the other definitions.

3.2.3 Plans and their execution

In order to make the user's goals come true in an environment, actions have to be taken. More specifically: Given a set of possible actions, the user's goal, and the current state, a (compound) action has to be identified that transforms the current state into a state contained in the user's goal. The basic mapping of Actions, goals, and states to a (compound) action is described by the following set P:

$$P == Actions \rightarrow SystemGoal \rightarrow SystemWorld \rightarrow Action$$

However, in order to get an action that really does what we want, we have to be more specific:

$$Planner : \mathbb{P}\, P$$

$$\forall\, p : Planner;\; pa : Actions;\; g : SystemGoal;\; e : SystemWorld;\; plan : Action \bullet$$
$$plan = p\ pa\ g\ e \Rightarrow$$
$$(plan\ e \in g) \wedge \quad \text{– Problem Solution}$$
$$(\exists\, ps : \mathrm{seq}\, pa \bullet plan = fold\ ps) \quad \text{– Sequencing}$$

$Planner$s are those functions in P that produce a (compound) action (the $plan$), which really solves the problem (which really transforms the initial state e into a state contained in the user's goal g) and which only use functions in the argument action set (pa) for creating the plan (such that $plan$ can be represented as a sequence[2] of actions taken from pa.

3.2.4 Putting it all together

In order to describe a complete system, we first isolate the assistant-function itself – this basically means to apply planning to a sequence of goals and to apply the resulting plans to the respective world states.

$$assist : Planner \rightarrow Actions \rightarrow \mathrm{seq}\, SystemWorld \rightarrow \mathrm{seq}\, SystemGoal \rightarrow \mathrm{seq}\, Action$$

$$\forall\, p : Planner;\; a : Actions;\; ws : \mathrm{seq}\, SystemWorld;\; gs : \mathrm{seq}\, SystemGoal \bullet$$
$$assist\ p\ a\ ws\ gs = frob\ ws\ (p\ a \circ gs)$$

The complete interaction cycle between user and environment is then described by the following function:

$$interact1 : User1 \rightarrow Planner \rightarrow Actions \rightarrow SystemWorld \rightarrow \mathrm{seq}\, SystemWorld$$

$$\forall\, u : User1;\; p : Planner;\; a : Actions;\; w_0 : SystemWorld \bullet$$
$$\exists\, gs : \mathrm{seq}\, SystemGoal;\; ws, ws' : \mathrm{seq}\, SystemWorld;\; as : \mathrm{seq}\, Action \bullet$$
$$ws' = \langle w_0 \rangle \frown ws \wedge$$
$$gs = u\ ws' \wedge$$
$$as = assist\ p\ a\ ws'\ gs \wedge$$
$$ws = frob\ ws'\ as \wedge$$
$$interact1\ u\ p\ a\ w_0 = ws$$

$interact1$ takes a user, a planner, a set of possible actions, and an initial state of the environment and maps these to the history of state changes that are created by the user's goals, the plans created for fulfilling the goals, and the state changes produced by applying the plans to the corresponding previous states.

gs is the sequence of goals produced by the user's will in response to the sequence of environment changes he observes, as is the sequence of compound actions (plans) that are created by the planner in response to the user's goals, and es is the sequence of environment states produced by consecutively applying the plans to the respective preceding environment states. Note that the user produces goals *in response* to environment changes and the environment is changed *in response* to the user's goals. So this definition intrinsically contains mixed initiative.

3.2.5 Observations

The following observations hold for the SAA-model as defined here:

- The complete model is based on just one given set – $SystemWorld$. This set is the root of the specification. Its precise definition is therefore the anchor point for further investigation.

[2] In fact, this linear order is a restriction, which is much stronger than required – especially when looking at the prospect of parallel execution of multiple plans. However, it is consistent with the concept of *one-copy serializability*, which is used as correctness criterion for distributed and replicated data management systems [2].

- Knowing the state of the environment is required at *all* steps of $interact1$ (this is the motivation for environment monitoring – i.e., context management[3].)
- The reference model – at this level – makes *no* statement whatsoever on which of these abstract functions are performed by the human user and which are performed by a machine.
- The current definition covers only *atomar* plan execution, which is in general not quite realistic.

3.3 Adding the real world

Until know, we have silently assumed that there is only *one* representation of the word and its states – and that system and user use this representation – $SystemWorld$ – for talking about the world.

Unfortunately, this view is somewhat too simple. The user does not formulate goals in terms of the *objective* state of the environment, but rather in terms of his *internal* model of this environment, the *mental model*. And the same holds for the system, which has its *own* model of the world.

In order to capture this situation, we first introduce the objective world, $World$, and the user's interpretation of this world, $UserWorld$:

$$[World]$$

$$[UserWorld]$$

(The system's interpretation of the objective world is $SystemWorld$.) Also, we add the mental model, which translates the objective environment by means of the user's cognitive system into its internal representation.

$$MentalModel == World \twoheadrightarrow UserWorld$$

Furthermore, the goals the user formulates are obviously defined in terms of this internal environment representation.

$$UserGoal == \mathbb{P}\ UserWorld$$

$$User == \mathrm{seq}\ UserWorld \rightarrow \mathrm{seq}\ UserGoal$$

The function for translating the objective environment into the system's view of the world is:

$$ContextModel == World \twoheadrightarrow SystemWorld$$

($ContextModel$ is the system's $MentalModel$.) $Actions$ have been introduced as transformations of the system's view of the world. In order to descibe the effect of actions in the real world, we obviously have to use the system's context model:

$$effect : ContextModel \rightarrow \mathrm{seq}\ World \rightarrow \mathrm{seq}\ Action \rightarrow \mathrm{seq}\ World$$

$$\begin{array}{l} \exists\ effect1 : ContextModel \rightarrow Action \rightarrow World \rightarrow World \bullet \\ \quad \forall\ c : ContextModel;\ ws : \mathrm{seq}\ World \bullet \\ \quad\quad effect1\ c = (\lambda\ a : Action \bullet c^{\sim} \circ a \circ c) \land \\ \quad\quad effect\ c\ ws = frob\ ws \circ (map \circ effect1)\ c \end{array}$$

The real work is done by $effect1$: it takes a real world, maps it to the system's world, applies the action, and then maps the result back to reality – this of course only the formal view ...

Finally, we can now define the "true" goals – which are defined by sets of allowable states of the real world:

$$Goal == \mathbb{P}\ World$$

[3]The notion of context applicable here is the one taken from mobile computing!

Also, we can describe how to translate user goals into system goals by means of $\mathit{MentalModel}$ and $\mathit{Context\text{-}Model}$:

$$\mathit{translate} : \mathit{MentalModel} \rightarrow \mathit{ContextModel} \rightarrow \mathit{UserGoal} \rightarrow \mathit{SystemGoal}$$

$$\forall\, m : \mathit{MentalModel};\ c : \mathit{ContextModel};\ g : \mathit{UserGoal} \bullet$$
$$\quad \mathit{translate}\ m\ c\ g = (\mu\, g' : \mathit{Goal} \mid g' = m^{\sim}(\!| g |\!) \bullet c(\!| g' |\!))$$

Before describing the new interaction process, we go one step further by looking at the human-computer interaction we have to consider.

3.4 Adding interaction

The model given in Section 3.2 above fixes the basic aspects of intentional human interaction with stateful environments. However, the distribution of work between the human and a potential (electronic) assistant is not yet defined (cf. the note in Section 3.2.2).

In the SAA-model, we have the following functional building blocks:

1. Environment observation and goal formulation
2. Planning
3. Plan execution (sequencing of individual actions)
4. Action execution (producing effects in reality)

We have now the choice of allocating these function blocks to either man or machine (or to a mixed control). If all blocks are assigned to the human, we have a proposal for a psychological model of cognitive action planning an execution. At the opposite extreme, we arrive at a model for autonomous robots.

In order to speak of *assistance*, at least one function block must be assigned to a human user and one to the machine. Also, one would prefer to allocate the "higher" functions to the human and the "lower" functions to the machine[4].

Using this approach, the highest possible level of assistance can be achieved by leaving only goal formulation to the user. We will call this "Class-1 assistance".

We now have to introduce an interface between user and machine at the goal-level: The user has to communicate goals to the machine. Unfortunately, goals are abstract entities whose internal representation in the human brain is of no help for communicating them to the machine, at least at the current state of the art.

Therefore, the user has to communicate his goal to the machine using a suitable external (physical) representation: A spoken sentence, a gesture, a sequence of keystrokes, etc. – basically a sequence of **utterances** of the user. And the machine has to reconstruct the abstract goal from the stream of utterances by such measures as speech recognition, gesture analysis, etc. – this process we will call **interpretation**.

First, we introduce the notion of user utterances:

$$[\mathit{Utterance}]$$

And we have a (conceptual) function ("built into the user") by which the user translates an abstract mental goal into a suitable utterance:

$$\mathit{Articulator} == \mathit{UserGoal} \rightarrow \mathit{Utterance}$$

A real user then is defined by three components: The user's will-function, his mental model of the world, and his way to express goals.

$$\mathit{user} : \mathit{User} \rightarrow \mathit{Articulator} \rightarrow \mathit{MentalModel} \rightarrow \operatorname{seq} \mathit{World} \rightarrow \operatorname{seq} \mathit{Utterance}$$

$$\forall\, u : \mathit{User};\ a : \mathit{Articulator};\ m : \mathit{MentalModel};\ ws : \operatorname{seq} \mathit{World} \bullet$$
$$\quad \mathit{user}\ u\ a\ m\ ws = a \circ u\ (m\ \circ ws)$$

[4] This does not always need to be the case. The SAMoA-Model [1, 6] allocates only task 3 to the machine, giving the user prompts to perform atomar actions not available to the machine.

And a suitable man-machine interface can then (conceptually) be constructed once we know the user's and the system's view of the world as well as the way the user utters goals:

$$mmi : Articulator \rightarrow MentalModel \rightarrow ContextModel \rightarrow \mathrm{seq}\ Utterance \rightarrow \mathrm{seq}\ SystemGoal$$

$$\forall\, a : Articulator;\ m : MentalModel;\ c : ContextModel \bullet \\ \quad mmi\ a\ m\ c = map\ (translate\ m\ c \circ a^{\sim})$$

4 Putting it all together

We have now all components we need in order to define the behavior of a complete Class-1 system:

$$system : Articulator \rightarrow MentalModel \rightarrow ContextModel \rightarrow Planner \rightarrow Actions \rightarrow \\ \quad \mathrm{seq}\ World \rightarrow \mathrm{seq}\ Utterance \rightarrow \mathrm{seq}\ World$$

$$\forall\, a : Articulator;\ m : MentalModel;\ c : ContextModel;\ p : Planner;\ t : Actions; \\ \quad ws' : \mathrm{seq}\ World;\ us : \mathrm{seq}\ Utterance \bullet \\ \quad \exists\, ws : \mathrm{seq}\ World;\ cs : \mathrm{seq}\ SystemWorld \mid ws = system\ a\ m\ c\ p\ t\ ws'\ us \bullet \\ \quad\quad cs = map\ c\ ws' \wedge \\ \quad\quad ws = (effect\ c\ ws' \circ assist\ p\ t\ cs \circ mmi\ a\ m\ c)\ us$$

The system consists of four components: the context manager $map\ c\ ws'$, the man-machine interface $mmi\ a\ m\ c$, the assistant kernel $assist\ p\ t\ cs$, and the actors $effect\ c\ ws'$.

Finally, in order to complete the definition of the SAA-model, we describe the interaction of a Class-1 system with the user and the external word:

$$interact : User \rightarrow Articulator \rightarrow MentalModel \rightarrow ContextModel \rightarrow Planner \rightarrow Actions \rightarrow \\ \quad World \rightarrow \mathrm{seq}\ World$$

$$\forall\, u : User;\ a : Articulator;\ m : MentalModel;\ c : ContextModel;\ p : Planner;\ t : Actions; \\ \quad w_0 : World \bullet \\ \quad \exists\, ws, ws' : \mathrm{seq}\ World;\ us : \mathrm{seq}\ Utterance \bullet \\ \quad\quad ws' = \langle w_0 \rangle \frown ws \wedge \\ \quad\quad us = user\ u\ a\ m\ ws' \wedge \\ \quad\quad ws = (system\ a\ m\ c\ p\ t\ ws')\ us$$

A sketch of the interaction between the various basic components of the SAA-model as defined by *interact* is given in Figure 2. This completes the definition of the SAA-model.

5 Conclusion and Outlook

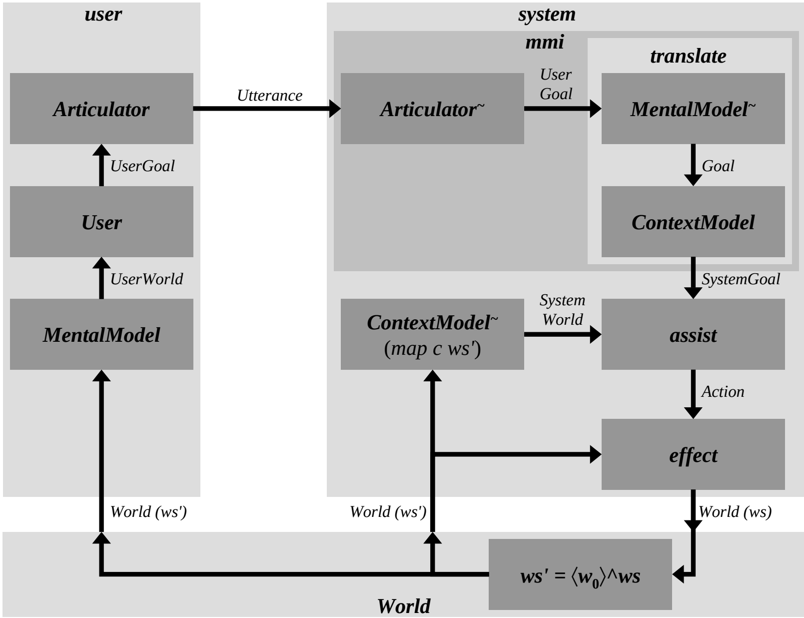
In this paper, we have introduced the concept of the "ecological interface" and we have outlined a reference model for capturing the essential entities whch are involved when looking at the communication processes that take place around this interface.

The SAA-model is currently beeing investigated as a conceptual foundation for the BMBF focus project "EMBASSI' (Electronic multimedia operating an service assistant) [4, 3]. Furthermore, it is th foundation of the SAMOA-Arcitecture under development at the Fraunhofer IGD [6].

The central aspects of the SAA-model are:

- Definition of a complete model, incorporating User, World, and System.
- Use of a goal-driven approach that allows the system to choose between different actions producing a comaparable effect.
- Clear separation between objective world (*World*), system view (*ContextModel*) and user view (*MentalModel*)

Figure 2 Interaction of fundamental components of the SAA-model, as defined by $interact$



- Conceptual inclusion of multimodal interaction ($Utterance$)

A specification of a concrete multi-component architecture for this model can be found in [7], where also an example implementation of this architecture based is given.

The current model obviously is a first draft and not yet complete. Specifically, it lacks the output branch (in order to describe bidirectional interaction between system and user). Also, modeling plans as atomar actions is a rather simplifid view, as errors and unexpected environment changes during plan execution need a more detailed discussion.

These points – and other refinements – are topics of ongoing work on the SAA-model.

References

[1] Chávez, E., Ide, R., Kirste, T. Interactive applications of personal situation-aware assistants. *Computers & Graphics*, 23(6):903–915, 1999.

[2] Coulouris, G., Dollimore, J., Kindberg, T. *Distributed Systems – Concepts and Designs*. Addison-Wesley, second edition, 1994.

[3] EMBASSI-Konsortium. MTI-Leitprojekt EMBASSI – Elektronische Multimediale Bedien- und Service-Assistenz. `www.embassi.de`, July 1999.

[4] Hildebrand, A., Sa, V. EMBASSI: Electronic Multimedia and Service Assistance. In *Proc. IMC'2000*, Rostock-Warnemünde, Germany, November 2000.

[5] Jia, X. ZTC: A Type Checker for Z Notation. `http://saturn.cs.depaul.edu/~fm/ztc.html`, August 1998.

[6] Kirste, T. Situation Aware Mobile Assistance. In *Proc. 1st Joint EC/NSF Advanced Research Workshop "Virtual Enviroments and Human-Centered Computing"*, Bonas, France, July 1–4 1999.

[7] Kirste, T. A reference model for situation-aware assistance. Forschungsbericht 01i002-FIGDR, Fraunhofer IGD Rostock, February 2001.

[8] MAP-Konsortium. MTI-Leitprojekt MAP – Multimedia Arbeitsplatz der Zukunft. `www.map21.de`, September 2000.

[9] Spivey, J. M. *The Z Notation*. Prentice Hall International, 2 edition, 1992. See also: `http://archive.comlab.ox.ac.uk/z.html`.