

Pillows as Adaptive Interfaces in Ambient Environments

Frank Nack^{1, 3, 4}, Thecla Schiphorst², Zeljko Obrenovic³, Michiel KauwATjoe⁴,
Simon de Bakker⁴, Angel Perez Rosillio⁵, Lora Aroyo⁵

¹Université Claude Bernard
Lyon 1
France

+33 (0)4.72.44.58.90

Frank.Nack_at_liris.cnrs.fr

²Simon Fraser University,
Canada,
thecla_at_sfu.ca

³CWI, The Netherlands,
zeljko.obrenovic_at_cwi.nl

⁴V2_, Institute for the Unstable
Media, The Netherlands,
name_at_v2.nl

⁵Eindhoven University of
Technology, The Netherlands
name_at_tue.nl

ABSTRACT

We have developed a set of small interactive throw pillows containing intelligent touch-sensing surfaces, in order to explore new ways to model the environment, participants, artefacts, and their interactions, in the context of expressive non-verbal interaction. We present the overall architecture of the environment, describing a model of the user, the interface (the interactive pillows and the devices it can interact with) and the context engine. We describe the representation and process modules of the context engine and demonstrate how they support real-time adaptation. We present an evaluation of the current prototype and conclude with plans for future work.

Categories and Subject Descriptors: H.5.2 [User Interfaces]: Haptic I/O, Input devices and strategies, Interaction styles, User-centered design

General Terms: Design, Human Factors

Keywords: Human-centred computing, Input devices and strategies, social interaction, haptic sensing, presence, tactile UIs, tangible UI, user experience design

1. INTRODUCTION

Designing ambient technology introduced into the potentially intimate personal space of a user is a complex problem, as a large range of design variables need to be addressed. Such technology inherently requires the ability to detect and understand the user's activity, body state and identity [2, 12]. A difficult and general problem with Ambient Intelligence is thus to translate low-level signals from the environment to meaningful environmental adaptation for the end-user.

Permission to make digital or hard copies of all or part of this work for personal or classroom use is granted without fee provided that copies are not made or distributed for profit or commercial advantage and that copies bear this notice and the full citation on the first page. To copy otherwise, or republish, to post on servers or to redistribute to lists, requires prior specific permission and/or a fee.

HCM'07, September 28, 2007, Augsburg, Bavaria, Germany.
Copyright 2007 ACM 978-1-59593-781-0/07/0009...\$5.00.

The aim of the work presented in this paper is to address this problem by exploring interfaces that go beyond direct manipulation. The interfaces we are interested in are based on everyday objects in adaptive ambient environments, such as the home or public, urban, and social spaces, i.e. lounges at cafes, bars. In this context adaptation is based on the input from a sensor-enhanced object with one or more low-level input streams, such as galvanic skin response (GSR), touch quality in form of pressure, or heartbeat. The environment tries to interpret this input, e.g. to identify excitement or detect an action, so that it can adapt itself in the most useful manner, e.g. adapting the environment to serve the mood or adapt the functionality of an object to support the currently performed task.

The key concepts investigated by our work include mobility, connectivity, invisibility, and configurability. We have developed a set of small interactive throw pillows containing intelligent touch-sensing surfaces, in order to explore new ways to model the environment, participants, artefacts, and their interactions, in the context of expressive non-verbal interaction. The long-term aim of our work is to establish a framework for ambient environments that is configurable by users either by integrating already established interaction and adaptation solutions for particular objects (such as the pillow) or designing new ones by themselves.

We first outline the motivation for choosing pillows as interfaces, based on a scenario situated in a living room. The main part of the paper provides an in-depth look at the technology used to turn a pillow into an interactive item according to the described scenario. We describe the structural layers of our system architecture and related technical detail. For providing the reader with an easy to grasp description of a pillow's abilities as well as of the feedback loop within the environment, we describe a subset of the living room scenario. Finally, we discuss the related body of work in the context of evaluating our approach and conclude with plans for future work.

2. MOTIVATION AND SCENARIOS

A problem with adaptive ambient environments is to interpret the **current user context** so that either appropriate **support of performed tasks** can be provided or the environment can be adapted to **improve or serve the user's mood**. In many cases it is both aspects that need to be addressed at the same time.

The aim of our work is to develop technology that facilitates environments to be adaptive to present parameters. We address four key issues, namely the representation of context, the selection of parameters that allow us to detect user actions as well as the user's mood, define efficient and non intrusive ways to collect those parameters and to define means to utilise these parameters for proper adaptation.

Context

Assume a user is in a private living room. There might be several interfaces running simultaneously, such as the TV, the stereo and the browser on the PC. Context-aware systems have to be aware of the user, interpreting hints, such as user position and actions performed, to find out which of these attracts the focus of the user's attention at the moment.

Context represents not only the material world, of which the user is a part, but also the user's cognitive state (i.e. attention, emotion). For example, the user might watch a program on TV and the system discovers a constant low attention rate. The reason might be that the program is bad but it could also be that the current light setting is irritating the user and thus distracts his or her attention.

The **first requirement** for our system is to represent the current state of the environment through descriptions of the various agents as well as the relations between them. The current state of the environment is therefore always a snapshot of the most recent event, which needs to be related to those events that already happened. A key issue is here the method to represent change.

What to Measure?

The two basic notions to be monitored in our approach are actions and cognitive states. Actions allow us to detect the user's focus of attention. The action performed immediately also tells something about the application utilised. Measurement of biometric data facilitates us to draw conclusions about the cognitive state of the user.

The **second requirement** of our system is to establish a key description set of

- user specificities, such as preferences for actions or strategies, likes and dislikes for content, etc,
- biometric data sets, e.g. containing touch efforts, GSR, heartbeat and methods of mapping those to high-level semantics.

How to Measure?

Collecting data, such as actions, is rather simple as technology is the general means to interact with audio-visual content. Logging this interaction is common practice. The collection of very personal data, such as GSR, pressure or heartbeat, is a sensitive matter. First, the required data needs to be reliable and thus feature extraction mechanisms on a meta level, such as the analysis of video surveillance, is by far not good enough for the time being. As we wish to support people in everyday environments it is also not suitable to treat them as guinea pigs in a laboratory by connecting them to sensors via cables.

The **third requirement** for our system is that taking measurements in an ambient environment needs sensors that collect biometric data. The sensors need to be made available through objects that invite the user's touch preferably for a longer time. Moreover, the object needs to be useable in private as well as public environments by a large

variety of user groups, i.e. children, teens, middle-aged adults and the elderly, and thus need to be also easily moveable between locations.

What to Adapt?

The reason for the adaptation of the environment is to either support the performance of the user in task processes with respect to the particular application the user is currently busy with or to adapt particular environmental parameters to improve or serve the user's cognitive state.

The fourth requirement for our system demands methods that map the results collected from the context descriptions to adaptive strategies, which address

- the currently used object itself or related objects depending on the task to be performed,
- environmental parameters, such as lighting or noise, for supporting, enhancing or calming the current cognitive state of the user.

Living Room Scenario

We developed an environment that addresses the outlined four requirements in a home scenario¹. The home scenario (see Figure 1) is based on the living room where the user consumes or interacts with different media sources. The environment is understood as an experience space, which the user should be able to explore freely without being overstrained or unchallenged. We utilize a set of small interactive throw pillows as the sensor-enhanced every day objects.



Figure 1: The living-room in the home scenario.

For example, the user might sit on the sofa, cuddling the preferred pillow, and watches a TV program. As the system has detected that the remote control is out of reach (context model) it has turned the surface of the technologically enhanced pillow into a remote control. The user makes frequent use of it, i.e. changing channels and uses the fast forward button. All that indicates a state of unexcitement, which is further confirmed through biometric data collected from the biometric sensors on the pillow. The action related data as well as the biometric feedback could now be used by

¹ We designed a second scenario, using the same framework, which is situated in a café/lounge environment, where participants are invited to re-mix a set of moving images projected in large scale on the walls within the café through the physical interaction with a set of the small interactive pillows. The focus of this work lies on an environment that is aware of users and objects but not necessarily knows much about them. This work integrates somatics and gesture interaction with textiles and interactive object design [16]. A description of this environment can be found in [5].

the adaptation engine to improve the excitement level of the user by suggesting to play a game. The result is another change of the functionality of the pillow surface, turning it into a play console. While playing the same data sources can be used to detect a level of cognitive awareness, which facilitates the environment to turn a part of the pillow into an alert interface, stating through soft vibrating and a text message, that the news channel on the PC provides interesting information related to the article the user was reading before she started relaxing in front of the TV. The aim of the home scenario is to explore the interaction between a single but well-known user and a single object interaction and its possibilities to influence the overall environment.

In the living room scenario there are a number of potential objects, such as chairs, sofas, teddy bears, blankets, pillows, watches or remote controls, that, enhanced with sensors and actuators, can serve as input and output devices. Yet, only a few fulfil the needs described in requirement 3.

Chairs and sofas are certainly touch-sensitive but their problem is their static nature. Teddy bears or alike would be excellent options with respect to their haptic sensibility. Yet, their user group is limited. Objects like a watch or the remote control are promising because they are hold or worn for a longer time. Yet, the way in which they are used is very limited.

Blankets and pillows are ideal candidates. Both are applicable in various surroundings and scenarios according to the user's needs. They can be found in the home environment as well as in public, urban spaces. More importantly, both evoke metaphors of intimacy and affection. Their surfaces of textiles or light-emitting material ask for touch. Finally, both can be, depending on its configuration, act as an input- as well as an output-device.

Our decision to use a pillow as a primary object of investigation was rather a question of pragmatics, as a blanket is too thin to hide batteries, boards, sensors, etc. Thus, for our prototype environment we decided for a pillow, of which a prototype is portrayed in Figure 2.

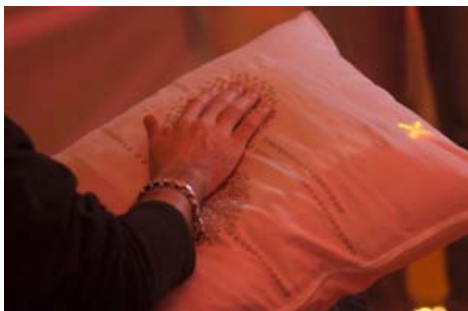


Figure 2. Pillow with actuators touchpad, LED display and vibrator.

3. MOVE.ME SENSORS, ADAPTATION AND FEEDBACK

In this section we introduce the technology developed for the pillow and the main technical developments of the *move.me* environment. The descriptive examples are taken from a subset of the implemented living room scenario. The subset scenario covers three states of a pillow, namely **sleeping state**, **game state** and **device**

state, as well as the transitions between them, i.e. **waking up**, **tune in** and **fading out**.

The main aspect of our prototypical ambient environment, called *move.me*, involves three parts:

- The user
- The interface, which contains the interactive pillow as input sensor and output devices, such as a vibrator, light-emitting diodes, screen, sound system, lamp.
- The context engine, which provides the analysis of the current state of the environment on which it might suggest an adaptation strategy, is a back-end server.

With respect to its interaction design our approach extends Don Norman's traditional execution-evaluation model [9] beyond the user's view of the interaction by including not only the interface but the entire elements necessary to judge the general usability of the interactive system as a whole. This allows placing the environment in different social contexts with an overlap on particular task.

As *move.me* is a reactive environment, we adapt for the context representation strategies from case-based reasoning (CBR) [1], in particular those strategies that trace the history of actions [7] to provide the means for a systems to adaptively interact with a user. Our real-time adaptation cycle is very similar to the reasoning process in CBR, which is often represented as a cycle composed out of five steps: elaborate (establish target case), retrieve (compare with existing cases), reuse (adapt the target case), revise (identify unspecified knowledge) and retain (establish new case in the case data base). We apply this cycle to the interpretation of sensor data. We start with a set of raw data, on which we then elaborate, based on user and environmental data (both together form our context), how to perform contextualized adaptation. The adaptation as well as the context history are then stored and are used in the ongoing process of ambient user adaptation.

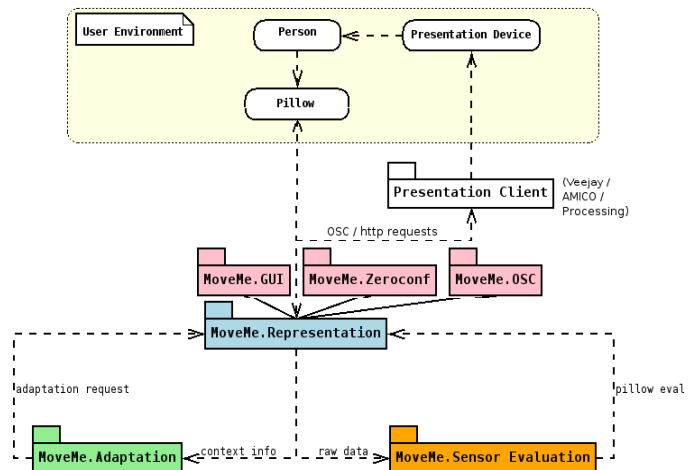


Figure 3: High-level architecture of the ambient environment, where the User environment box represents the room with client (e.g. the pillow) and everything outside this box represents the server or central system.

In the remaining part of the section we outline the various modules of the *move.me* environment, as portrayed in Figure 3, namely:

- *Sensors*, i.e. the input sensors of the pillow

- *Sensor evaluation module*, which instantiates the device drivers for every detected pillow and evaluates incoming raw data. Its main task is to perform some statistical analysis (some low-pass filtering and mean value calculations) in order to keep the overhead of processing load low (the Sensor Evaluation box at the bottom right in Figure 3)
- *Context module* that consists of data structures describing the current context with respect to users, devices and the interactions between them. (the Representation box in the lower middle of Figure 3)
- *Communicator module*, (the presentation client as well as the Zeroconf and OSC box in the middle of Figure 3) which enables the flexible connection of all components, and provides interfaces towards external systems, such as a the TV, a laptop, the stereo, the light sources, etc.
- *Adaptation engine*, which uses data from the context module to establish a mapping between detected action and the appropriate environment adaptation. It also conveys instructions about the source to be adapted and the means of adaptation to the other components (the Application box at the bottom left in Figure 3).

The central system (sensor evaluation module, context module, and the adaptation engine) is developed in Python (ver. 2.4) and runs on GNU Linux and on Apple OSX 10.4 in combination with Fink. The Communicator part of the system is implemented as a Java application based on the AMICO (Adaptive Multi-Interface COmmunicator) framework (<http://amico.sourceforge.net/>).

In the following sections we describe the various modules in more detail.

3.1 Pillow Sensors

A pillow in our environment is the main medium for collecting sensor data, and providing limited feedback to the user. A pillow can be equipped with any of the sensors and actuators displayed in Figure 4.

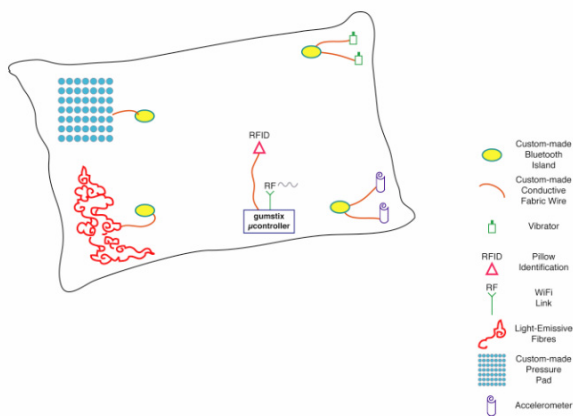


Figure 4: Pillow technology

Figure 5 portrays a particular pillow prototype, equipped with a touchpad (large rectangle on the pillow in Figure 5), a LED display

(smaller rectangle below the touchpad) a vibrator (the little round box to the bottom left of the pillow) and a Gumstix² Connex Linux computer expanded with a SIOS (Sensor Input Output System) daughter board plus the RFDI reader (the part left of the pillow). This prototype is shown without a cover. Complete prototypes with cover are shown in Figure 6.

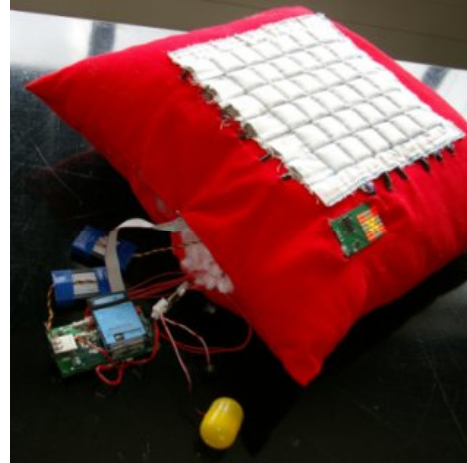


Figure 5: Functional pillow without pillow cover, exposing the technical components



Figure 6: Fully functional prototype pillows with covers

The main sensor we apply is a touch-based interface for measuring pressure. Bleepers, vibrators and light sources, either in form of light emitting fibres or as a LED display, provide localized feedback.

The touch-pad itself is a simple grid (64 sensitive regions) of carbon impregnated open cell polyethylene foam. This material exploits the characteristic that electrical resistance of the foam drops as the density of the foam increases. We utilize this behaviour to identify a point of contact or applied pressure. We exploit this behaviour to identify a point of contact or applied pressure. I2C is used to read and steer the pressure sensor circuitry. To control the Led display we use the SPI Bus. Please note that the current pillows connect the various hardware parts via wires instead of the bluetooth islands as suggested in Figure 4.

The processing unit located inside the pillow is a Gumstix³ Connex Linux computer expanded with a SIOS (Sensor Input Output System) daughter board developed by the V2_lab. The Gumstix filters the incoming pressure data and communicates these in discrete packages to the server application. The communication between pillow and server uses the OpenSoundControl (OSC) protocol. We employ OSC because it is a simple protocol widely used by other applications. Thus, using OSC enables us to talk

² Gumstix is a registered trademark by Gumstix inc.; <http://www.gumstix.com>

³ Gumstix is a registered trademark by Gumstix inc.; <http://www.gumstix.com>

easily with many other platforms, as long as both sides agree over the semantics of messages send. When switched on the pillow automatically propagates itself on the network, sending its IP-address and port number, using a combination of Zeroconf networking (<http://www.zeroconf.org/>) and an OSC Querying System. This allows listening programs to detect and register the pillow and start receiving or sending messages.

For identifying users in the closer surroundings of the pillow each pillow contains an RFID reader (SonMicro CY8C0105-B5 RFID Module). The module also acts as a Writer, allowing us to include personal codes into programmable RFID-Tags, which users of the environment wear as bracelets or which are via small tags embedded within each pillow.

Emphasizing the sensual aesthetic of a pillow, covers are designed to encourage connection in an associative and intuitive way. Pillowcases are made out of conductive fabric textiles, such as silk organza [15], which allow qualitative recognition of touch on the surface of the pillow. They are also transparent enough so that icons displayed on an LED are still visible. Pillow prototypes are portrayed in Figure 6.

Pressure is the essential type of data we process to extract a caress and its effort. We have identified a set of parameters that can be extracted or calculated from the information that the response area provides over time. At the moment we utilise the following pressure input parameters: **touch intensity**, **size** (the size of the interaction object that touches the pad), the **speed** of the touch and the **direction** of the touch.

3.2 Data Evaluation and Abstraction

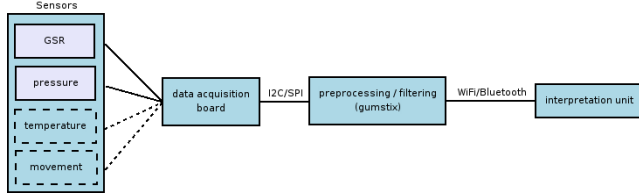


Figure 7: Schematic overview of dataflow from the sensors to the interpretation unit in the central system

The pillow itself only provides a basic filtering of the signal which is then transmitted, as described in Figure 7, along with additional information, such as which other pillows or users are near this pillow, to the central system, where all the other processing happens (e.g. calculating time and space variables). The transition rate for the data of the differently sized pads available (3x3, 6x6, and 8x8) is between 20 to 32 Hz.

Once the data for pressure, size, speed and direction is received by the central system the sensor evaluation module performs a first abstraction on the data. The way that incoming sensor data is analyzed depends on the context and the configuration of the pillow. For example, if the pillow contains a pad that allows measuring pressure and an accelerometer for computing movement a different abstraction scheme is provided as output compared to a pillow that only contains a pressure pad. This means that the system treats each individual pillow configuration/setting differently, as it also does for users. Only at a later stage the individual views are combined to the global context view.

When analysing streams of data, here the input data for every taxel of the pad (a taxel is one of the different tactile grid cells), it is

desirable to keep a history of past events especially when we are looking for trends in the data [11]. However, storing the whole dataset and re-iterating over the last n samples whenever a new sample arrives quickly becomes inefficient as n grows. We, therefore, apply a method that keeps a history of past events without actually storing them but summarizing instead the entire set of (or the last n) past events in a few critical parameters. If we calculate the mean, standard-deviation, minimum and maximum value over the past events, we can compare the value of the current event with a set of threshold values generated from the history of past events. Incoming values are first compared with the current set of thresholds, and then they are used to update them. Thus for every taxel of the pad we can calculate the current state based on the following calculations:

output	condition	meaning
0	$x < \min$	x is lower than ever before
1	$\min < x < (\text{mean}-\text{sd})$	x is very low
2	$(\text{mean}-\text{sd}) < x < \text{mean}$	x is lower than average
3	$\text{mean} < x < (\text{mean}+\text{sd})$	x is higher than average
4	$(\text{mean}+\text{sd}) < x < \max$	x is very high
5	$x > \max$	x is higher than ever before.

where X represents the current input signal from the pillow on the taxel and the output value (output) represents the type change to the taxel since the last measurement.

During our evaluation sessions (see Section 4 for details) we found, however, that a pure threshold-based approach often leads to unpredictable pillow behaviour.

We therefore introduced an additional evaluation step, which is based on the idea of utilising the Centre of Gravity (or Centre of Mass) to determine the average pressure point location of the touch pad. Giving every cell a mass proportional to the pressure applied (we define the cells variance as its mass) we can compute the COG as the average of the cells positions in the matrix weighted by their masses. If there is no change in action the COG converges to the absolute centre, which is simply the average of all locations weighted by 1. We use this property to determine the start and end of events. From the COG we also derive direction and speed.

Although this approach proves to be more flexible and lends itself for a more precise fine-tuning of the detection of action and its location, at this point we cannot distinct (using COG alone) between a positive (pressed) or a negative (released) event. If a pressure is sustained for long enough the COG will converge to the absolute centre and a release will trigger a new event. Thus, at the moment we simply have more events to cope with.

The results of the sensor evaluation provide a localized event description, as changes can be related to taxels, in form of a matrix that represents all taxels of the touch pad, where every taxel is identified as started, running or ended (a sort of second order of gravity). This description of position and event status is send to the corresponding session buffer of the context module, where the next level of abstraction on the results is performed.

3.3 Context representation

The **Context Module** describes the current (present) status of the environment with respect to resident users, devices and the interactions between them.

User as well as device profiles are loaded into the Context Model once their RFID is detected. For each detected user or object a memory structure is established (**Context User Model (CUM)** and **Context Device Model (CDM)**), which reflects only those characteristics that are relevant for the current context.

User characteristics are, for example, the user identifier, the current biometric status, current action, current interest (i.e. leisure as derived from the playing mode of the device mainly used at the moment), current location, as well as descriptions of likes and dislikes with respect to the potential media to be consumed and the related presentation devices.

Device characteristics include the device IP, its input sensor set, its actuator setup, current activity, current state for every input sensor and output actuator, its current location, and particular behaviour descriptions, such as adaptability with respect to input sensors, where adaptation here, for example, means automatic functionality changes of regions on a touchpad). Note, it is the sensor set and the behaviour descriptions that trigger the instantiation of the device drivers and are thus partially responsible for the type of sensor evaluation. For example if a pillow is in the 'device state' the touchpad of the pillow might be organised as described in Figure 8.

	0	1	2	3	4	5	
Top left	Top left			Top right	Top right		0
Top left	Top left			Top right	Top right		1
		Center	Center				2
		Center	Center				3
Bottom left	Bottom left			Bottom right	Bottom right		4
Bottom left	Bottom left			Bottom right	Bottom right		5

Figure 8: Region pattern for a 'device state' of a touch pad

Depending on the associated role of the device, e.g. the touchpad functions as a remote control, the provided events for the region (see section 3.2) will be interpreted differently.

The reason why we provide devices with similar description structures as those for users is so that devices themselves can become proactive towards users. At the moment we only store simple data, such as user id, action performed and its duration but we wish to explore further in that direction (see section future work). It is also important to mention that the CUM and CDM are subsets of the general user or device model, which contain the overall description of the user or device. The user model might also contain descriptions of relations to friends, which might not be applicable for the current context of watching TV alone in the living room.

An instance of a CUM or CDM will be deleted from the context model in the very moment the related agent (user or object) is not part of the context any longer. A reason might be that the user left the room or that an object is switched off or removed from the room. In that case the context module also updates the general user or device model.

An appropriate representation of the current status of the environment requires that the model is constantly refreshed. We have noticed during our user tests (see section 4. Evaluation) that the average adaptation rate that causes a latency of around 0.2

seconds is not perceived as slow by users. A quicker adaptation call-back would put unnecessary load on the computer system, and would not produce better user experience.

Yet, both CUM and CDM are rather static schema, providing a current situational "snapshot", with which it is difficult to observe and track the dynamics of the environment. We introduced, therefore, the concept of a session, which monitors the interactions between a user and a device or between devices over time. A session is a data structure containing the identifications of the agents involved in the interaction, the start/end time of the session, the recorded sensor data, and resulting derivations such as type of pillow activation, pillow state and user state.

Thus, a session provides a unified view on a series of interaction events exchanged among agents. As long as a user interacts with a pillow, the session records all detected changes of user and pillow and the relations between these changes, e.g. which action influenced which change. The session description is in a way a 'case' composed out of events. At the moment the current event and the recent history of events are used by the adaptation engine to determine user state changes, which are used to propose adaptation strategies if required. In the future we would like to post process session description for further automatic improvements of the user model as well as improvements of the adaptation behaviour.

A session is instantiated by the Context Module once a device detects the ID of the user and the pillow sensors register relevant data. For example, a use case in which the user utilizes the pillow to control a presentation of an audio system is described by two sessions: the user interacting with a pillow (session A: userID – pillowID) and the pillow interacting with the audio system (session B: pillowID – AmplifierID). Sessions are closed, either once the interaction stops or when the user leaves the context. Why is this distinction relevant? There are cases where several users might share a device, e.g. several users touch a pillow. In such a case, where more than one user is detected by a device, the one using the device the longest is considered the prime user. If the prime user leaves the context, the session is terminated. Simultaneously a new session is opened, though, where the user being second in the list will automatically assigned prime user. It is at the device to indicate this change to the remaining users. Once a session is terminated it is stored in the History Model.

The **History Model** is our approach towards an individualized long-term memory of the interaction patterns for every user and device. It is updated once a session has been terminated. The model contains at the moment two data sets, namely **identification** [userid or deviceid / context type / indate / outdate] and **session** [agent1 / agent2 / context type / actionlist / datalist / adaptationlist]. The identification set is automatically instantiated when a user enters a context. This set serves as cross check source for the adaptation engine to evaluate user behaviour. The session set describes every interaction the user or device was involved in (pattern, duration). The data set stores the collected biometric data (thresholds, duration) and the adaptation list contains the adaptations performed by the adaptation engine based on the data in the same time frame (adaptation method, success value, duration). At the moment we only keep track of sessions, and thus make them accessible to the adaptation engine, in form of a sqlite2 SQL database. Real instant or post evaluations (e.g. at the end of a day, week, month, etc) still need to be developed.

The outlined representation structures serve as input sources for the adaptation engine to determine if an adaptation is required and in case it is, which type of adaptation needs to be performed.

Before we explain the adaptation engine we first briefly describe the *Communicator module*, (see Figure 3) as it serve as the output source for the adaptation engine.

3.4 Communication between objects

As the pillow can be used as an output device the environment needs to be able to easily establish a connection between the pillow and the presentation device it might be connected to. For connecting the pillow server with other multimedia component, we use the Adaptable Multi-Interface COmmunicator (AMICO) AMICO is a generic platform, used to support rapid prototyping with OSS components in different domains [reference removed for double blind review]. The proposed brokering infrastructure is based on the publish-subscribe design pattern. It is well suited for integration of loosely-coupled parties, and often used in context-aware and collaborative computing. When using simple data structures, the loosely coupled approach can be highly adaptable, so that new applications can both reuse existing data in the repository and add their own data without breaking the infrastructure. This approach is also fault tolerant, as components run as independent processes. In the loosely coupled model, components can run on different machines in a distributed environment. Components communicate by exchanging events through a shared data repository consisting of named slots. Components can update the slots, and register for notifications about changes.

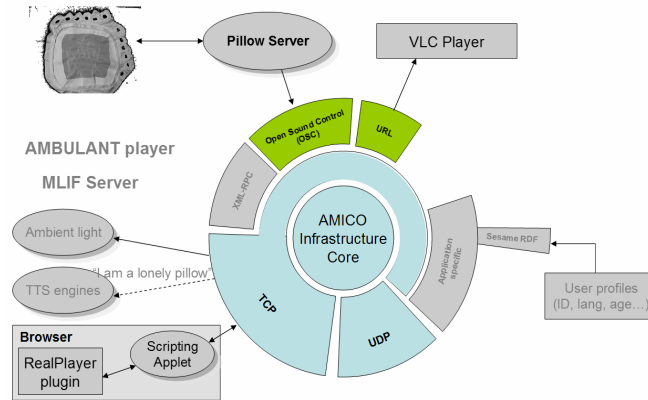


Figure 9: Configurations of AMICO middleware for integration of the pillow server with other multimedia components.

Figure 9 illustrates how this infrastructure is used to connect the pillow server with other multimedia components. We have integrated the pillow server with the infrastructure through the OSC interface, which is also used for the communication between the pillow's hardware and its driver software. AMICO uses the outcome of the adaptation engine, namely commands for the multimedia players⁴, controlling sound intensity and playback, as well as

⁴ We used several multimedia players, included VLC player, connected through HTTP interface, the AMULANT SMIL player, connected through XML-RPC interface, and RealPlayer player embedded within Web page and connected with out infrastructure.

sending a message to the user, which have been generated on the basis of the discrete actions provided by the pillow.

3.5 Adaptation and Feedback

In our environment adaptation focuses on three major processes, namely **stimulation**, **relaxation** and **representation**. *Stimulation* describes the attempt to either engage a non-active user into an interaction with the environment or to increase a low-base activity. *Relaxation* tries to reduce the amount of activity or excitement. *Representation* aims to present the state of the environment and the user in a visible and audible form, to give the user a feedback about actions and their effects. The adaptation engine, therefore, constantly evaluates data from the Context Module, and reacts on changes only if they are outside the provided constraint set.

The adaptation process is based on the idea of a finite state machine (FSM), a model of behaviour composed of **states**, **transitions** and **actions**. The states are defined by CUM and CMD descriptions, and the session structure from the Context Module. The transitions are based on the constraints set by the administrator, which, in case of our home scenario, can be everybody in the house. At the moment we implemented a system with a single constraint set for a room but the aim is to later provide constraint sets applicable for every single individual.

We employ two types of constraints. The first type covers excitement levels, ensuring that a certain upper or lower excitement threshold is not crossed. The other type of constraints are time constraints, which describe for example the duration in seconds the system should wait to suggest new activities or the time interval for verifying that a suggested action is indeed performed by a user. These constraints are set by a user and become part of the user model, in the preference section. This is one of the possibilities of the user to control the system, which enforces a certain trust in its capabilities to adapt.

If a constraint cannot be fulfilled, the adaptation engine utilises actions to ensure, for example, that the tolerated excitement levels is reached again within the temporal constraints defined by the user⁵. Actions describe the adaptation performed at a given moment. An action features an instruction for the behaviour of an object, e.g. instructions for the vibrator of a pillow to vibrate in a particular pattern suggesting a particular information code. An action might also trigger a number of events at the same time, such as changing the TV channel.

Actions are organised in form of rules and context scripts. The organisation of rules is based on context scripts. In our environment the scripts are designed by us and thus represent an already established interaction and adaptation solutions for a particular object, namely the pillow. As the language for the rule set is simple, it is, however, also feasible to assume that users can improve or enhance the behaviour set or designing new ones for different objects.

For a single user single object interaction the knowledge base is rather simple. Elsewhere [reference removed for double-blind review] we described that this approach also works well for more

⁵ It is the user who determines the actual meaning of the thresholds, where the system only reacts on them. This reduces the inference level of the system greatly.

complex settings, such as interaction between a group of people and one object, a group of people and several objects, etc.

The general structure of rules in *move.me* is as follows:

```
<proposition> ::= if <input-statement> then <output-statement>
<proposition> ::= if <input-statement> then <output-statement>
                    else <output-statement>
```

An input statements can consist of an input fact, a negated input fact or multiple nested input statements. Input facts correspond to the states that result from the sensor evaluation module. They might look as follows:

```
<input-fact> ::= <corner-touch>
<input-fact> ::= <c1-touch>
<input-fact> ::= <not-pressed>
```

Output facts correspond to a particular device, i.e. the LED, the vibrator, etc. as well as to the commands to be sent to the AMICO presentation client. Note, a naming synchronization between AMICO and our environment is necessary so that the actions can be performed accordingly. Output facts for animating an icon might look like this:

```
<output-fact> ::= <show-play>
<output-fact> ::= <blink-wait>
<output-fact> ::= <animate-icon-7>
```

where show triggers a single icon, blink repeatedly turns it on and of with short intervals, and animate displays a predetermined sequence of related icons. Note, a set of so far 30 icons is preloaded into the sensor board.

Actuators, such as the vibrator or a loudspeaker, provide parameters related to their modality space. Examples for a vibrator (buzzer) or loudspeaker or look like this:

```
<output-fact> ::= <buzz-soft>
<output-fact> ::= <buzz-gradual-increase>
<output-fact> ::= <beep-note-G3>
<output-fact> ::= <beep-increasingly-louder_C1>
```

The three parameters are flexible enough to be configured endlessly and thus it is the design of the rules themselves that provides the sense-making. It lies in the hand of the user to determine the behaviour of the environment, which adds to its trustworthiness.

Let us describe the adaptation mechanism based on the state model portrayed in Figure 10, which represents the pillow states **sleeping**, **game** and **device**, as well as the transitions between them, i.e. **waking up**, **tune in** and **fading out**.

We already mentioned that the adaptation engine constantly evaluates data from the Context Module, and reacts on change. Assume that the system detects that a user enters the room (detection of the RFID tag). The context will load the user data relevant for the scenario (i.e. living room) and wait for new interaction sessions. If none is established within the time threshold for inactivity provided by the user parameters the adaptation will try to engage the user into an activity. All actions to be executed here fall into the *stimulation* category.

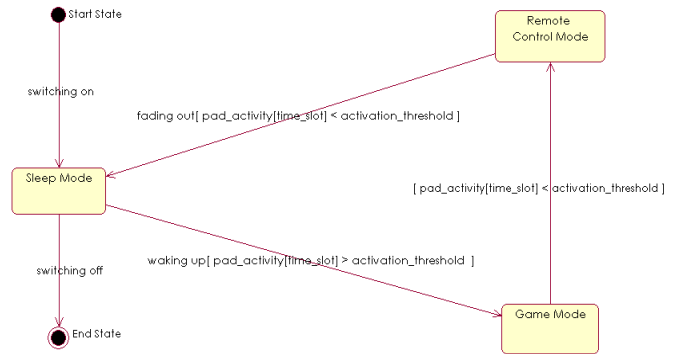


Figure 10: Potential pillow states and transitions between them.

‘Stimulate user’ activates a list of action scripts, of which one might cover attracting the user to the nearest pillow. As a result the adaptation engine will send a command to the closest pillow (location information taken from the device model and the user model) via AMICO as shown in the result slot of the rule below:

precondition	result
<sleep-state>	<purr> every 30s
	<show-asleep> every 30s
	<amico-sleep-mode>

This rule stands for: let the pillow make a gentle purring sound (a buzzing pattern), show a pattern of z’s on the LED and let that continuously run.

The adaptation engine observes the context model for signs of contact, i.e. the standard deviation of sensory data rises above a given threshold. If that is not the case within a given time span then the adaptation engine would try another strategy, which means utilize a different action provided by the stimulation set. Note in such a case the system would store the given context, e.g. type of category (stimulate), user id, strategy chosen, and the result (negative) in the adaptation part of the history model. For the rest of the example let us assume that the user responds according to the provided strategy.

After the user has started to interact with the pillow, the adaptation engine will first notify the user that this interaction is recognized (category: representation), as illustrated with the following script:

precondition	result
<sleep-state> and (standard_deviation > threshold) for > 0s	<transition-buzz>
	<show-loading>
	<amico-waking-up-mode>

where <transition-buzz> is a short sequence of buzzes identical for each transition. <show-loading> displays an animated icon that slowly fills the display with burning LEDs. Internally the state of the pillow is now set ‘waking-up’.

Realising that a session is instantiated the adaptation now attempts to instantiate an activity between user and pillow. The user activity list of the user model describes the preferred action states of the user, e.g. TV, Game, PC, etc. Depending on the rank within the list as well as the last state performed during the last session of the time the user was present in the context (living room), the adaptation engine picks a state, for example the game state. Thus, it sends a command to AMICO requesting to show the game icon on the pillow.

The adaptation engine now waits for a particular time (not more than 5 seconds) and checks if the standard deviation of the relevant

cell on the touchpad changes, meaning that the user acknowledged that she wishes to play. As a result the engine instantiates a “go” icon (category: *representation*) and starts a game accordingly. Note, the device model is now updated that the pillow acts as a game console of type synthesizer.

We have implemented a little sound synthesizer the user can play. The sound is a function of the center of gravity, acceleration, speed, the percentage of taxels that are pressed, and the intensity (the percentage of total pressure intensity). We are not playing actual frequencies but notes. The audio output is a simple Pulse Width Modulation circuit. In that case the adaptation engine performs the task of a game engine as well as the observer of the user. Once the adaptation engine detects low activity on the touchpad, it tries to initiate a new stimulation by suggesting an action change from Game to TV.

The user can now use the touchpad now as a remote control, where the functionality of the cells might be as such:

active fact	result
<clockwise-movement> and length > n	<amico-play-movie-next> <show-next>
<clockwise-movement> and length > n	<amico-play-movie-prev> <show-previous>
<c2-touch>	<amico-fullscreen> <show-fullscreen>
<c3-touch>	<amico-volume-up> <show-volume-up>
<c4-touch>	<amico-volume-down> <show-volume-down>
<c1-touch> and <c2-touch>	<amico-pause> <show-pause>

The synchronization between the pillow remote control and the TV is completely handled by AMICO. The adaptation engine now investigates the two established sessions, namely user and pillow and remote control and TV set. The main input will nevertheless be from the pillow, as this provides the biometric data from the user.

If the user leaves the context, which the adaptation engine realizes through closed sessions and the disappearance of the user id from the list of present users, then the adaptation engine applies a pillow transformation (category: *representation*) by letting the pillow fade out of the remote control state into ‘sleep state’.

precondition	result
<remote-control-state> and <number-of-events> > 0 and (standard_deviation < threshold) for > 30s	<showunloading> <amico-fading-out-mode> <transition-buzz>

postcondition	result
<remote-control-state> for > 30s	<sleep-state>

If the adaptation engine does not detect any action, i.e. the standard derivation remains 0 for a defined amount of time or there are no users in the list of present users for a defined amount of time, then it sets all registered objects into state of stand-by.

4. EVALUATION

We performed one elicitation study on the initial prototype, as presented in Figure 2, as well as a test at the demo event of the funding agency that supports this project, where participants had the chance to experiment with a set of pillows as described in Figure 1, 6 and 7..

The qualitative elicitation study [5], took the form of a one-day participatory workshop with 10 users (3 females, 7 males), and covered:

- A hands-on free exploration session with a medium-fidelity pillow prototype.
- A “Wizard of Oz” simulating the complete intended functionality of the system.

The participants’ experiences during both sessions were video taped and later analysed.

Apart from showing that the current architecture is stable and functional, especially in terms of mobility and connectivity, the workshop provided two major findings. First, the initial interactive exploration of the pillow lead to the users’ full engagement with the system even when the feedback in form of iconic messages on the pillow’s LED display was slow. Second, in the interview session, users indicated that they would appreciate the idea of device memories and the long-term memory of the system, even though we where not be able to test these elements during the workshop.

The presentation of the pillow functionality at the symposium showed that the adaptation based on stimulation, relaxation and representation is sufficiently sound to let users understand how the environment reacts on their actions without explicit training session. Moreover, it could be demonstrated that the rule set can be adapted on the fly, even though the coding was done by one of our group members based on the wishes of the visitors.

Because both cases our test sample are rather small, we were only able to perform a qualitative evaluation of the system. However, we have taken these findings as general guidelines, which will allow us to make educated decisions for further developments.

5. RELATED WORK

There has been a great deal of general research in sensing and biofeedback in human-computer interaction [3, 15] This research has indicated that a number of well established sensory methods, such as pressure and GSR, can be efficiently reused in our context to obtain a window into the state of an individual. In addition, there is work that illustrates the elevation of low-level signals to higher-level interaction [14].

The affectionate quality of a pillow is also used in other works, such as the interactive pillows by Ernevi⁶ et al. [10] and Aoki et al. [4]. Both projects investigate interactive pillows as a means of enhancing long-distance communication. We share the notion of a sensual aesthetic of a pillow that encourages connection through feel in an associative and intuitive way, we do support a different vocabulary of expressiveness, based on action.

Buxton et al [6] provides early descriptions of the unique characteristics of touch tablets relative to other input devices such as mice and trackballs. Chen et al [8] describe the use of a touch-sensitive tablet to control a dynamic particle simulation using finger

⁶ <http://www.tii.se/reform/projects/itextile/pillow.html>

strokes and whole-hand gestures, where the gestures are interpreted as a form of command language for direct manipulation. In our work we go beyond direct manipulation, enriching the interaction with parameters about the quality efforts.

The authors in [16] describe the use of kinaesthetic motion-analysis models to represent movements. The current work was influenced by this choreographic approach to motion description and presentation of these studies.

Haptics and touch have been explored by many researches. The University of Tsukuba, for example, has developed a large number system that makes use of haptics such as finger/hand manipulation and locomotion [13]. Although these systems use different technologies, they have provided us with a motivation for usage of touch.

6. CONCLUSION AND FUTURE WORK

In this paper we have described an ambient environment in which a set of small interactive throw pillows containing intelligent touch-sensing surfaces allow users to change the behaviour of the system based on the actions performed by the user. We have demonstrated that pillows serve well as adaptive interfaces in changing contexts and described a novel framework that facilitates the integration of various devices to enhance the adaptation of the environment.

A novel aspect of *move.me* is the approach to map efforts of users actions to higher-level adaptation activities, by defining the mapping space between biometric data and its potential meaning. Although the first prototype shows promising results, we have to provide significant improvements with respect to adaptation response time as well as the range of adaptations to facilitate a richer experience environment that better reflects the motions of the social interchange.

Fruitful future research aspects are how different modality sensors, e.g. pressure and galvanic skin response can be combined into one adaptation framework. Second, we want to experiment with different forms of archival and retrieval of user experiences during interaction.

7. ACKNOWLEDGEMENTS

This work was supported by the ITEA Passepartout project. We wish to thank Philips for supporting the students of the *move.me* project. We also express our gratitude to V2_ for having established the technological environment that allowed the development of the *move.me* system.

8. REFERENCES

- [1] Amodt, A., & Plaza, E. (1994). Case-based reasoning: Foundational issues, methodological variations, and system approaches. *AI Communications*, 7, 39-59.
- [2] Aarts E., "Ambient Intelligence Drives Open Innovation", *interactions*, July-August 2005, pp. 66-68.
- [3] Andreassi, J. L., - *Psychophysiology (Human Behavior & Physiological Response)*. Fourth Edition, Lawrence Erlbaum Associates Inc, 2000
- [4] Aoki, K. and Kamada, T. : Sanekazura, 2005. http://www.ok.sfc.keio.ac.jp/orf/2005/en/sanekazura_en.html
- [5] Aroyo, L. et al: "Personalized Ambient Media Experience: *move.me* Case Study", Submitted to International Conference on Intelligent User Interfaces (IUI) 2007, Hawaii, January 28th – 31st. 2006.
- [6] Buxton, W.: *Gesture and Marking*. Chapter 7 in Baecker, R.M., Grudin, J., Buxton, W. and Greenberg, S. (Eds.), *Readings in Human Computer Interaction: Toward the Year 2000*, Morgan Kaufmann Publishers, San Francisco, 1995.
- [7] Champin, P.A., Mille, A., and Prié, Y.. "MUSSETTE: Modelling USEs and Tasks for Tracing Experience." ICCBR'03 : Workshop "From structured cases to unstructured problem solving episodes" ICCBR'03: NTNU, 2003. 279-286.
- [8] Chen, T.T.H., Fels, S.S. and Min, S.S.: *Flowfield and Beyond: Applying Pressure-Sensitive Multi-Point Touchpad Interaction*, Proceedings IEEE Intl Conf on Multimedia and Expo, Baltimore, 2003.
- [9] Dix, A., Finlay, J., Abowd, G., and Beale, R. 2004. *Human Computer Interaction*. Hemel Hempstead: Prentice Hall International (UK).
- [10] Ernevi, A., et al. *Tic Tac Textiles*. In: Redström, J., Redström, M. and Mazé, R. (eds). *IT+Textiles*. IT Press/Edita Publishing, Helsinki, 2005
- [11] Fayyad, U. M., Gregory Piatetsky-Shapiro, G., and Smyth, P.: *From Data Mining to Knowledge Discovery in Databases*. *AI Magazine* 17(3): 37-54 (1996)
- [12] Goleman, D. *Emotional Intelligence*. Bantam Books, New York, NY, USA, 1995.
- [13] Iwata, Hiroo "Art and technology in interface devices", Proceedings of the ACM symposium on Virtual reality software and technology VRST '05, Monterey, CA, USA, 2005, pp. 1-7.
- [14] Kerhervé, B; Ouali, A and Landon, P.: *Design and Production of New Media Artworks* Proceedings of the 1st ACM MM 05 Workshop on Multimedia for Human Communication - From Capture to Convey (MHC 05), pp. 18 - 22, November 11th 2005, Hilton Hotel, Singapore.
- [15] Picard, R. W. *Affective Computing*, The MIT Press, 2000.
- [16] Schiphorst, T., Jaffe, N. and Lovell, R.: "Threads of Recognition: Using Touch as Input with Directionally Conductive Fabric", Proceedings of the SIGCHI conference on Human Factors in computing systems, April 2–7, 2005, Portland, Oregon, USA.