

A cognitive model of user interaction as a guideline for designing novel interfaces

Felipe Aguilera¹, Rosa A. Alarcón², Luis A. Guerrero¹, César A. Collazos³

¹ Department of Computer Science, Universidad de Chile
Av. Blanco Encalada 2120, Santiago, Chile
{faguiler, luguerre}@dcc.uchile.cl

² Department of Computer Science,
Pontificia Universidad Católica de Chile.
Av. Vicuña Mackenna 4860, 6904411, Santiago Chile.
ralarcon@ing.puc.cl

³ System Department, Universidad del Cauca
FIET, Sector Tulcan, Popayán-Colombia
ccollazo@unicauca.edu.co

Abstract. Adaptive systems behavior based on user models appear promising, mostly for complex environments such as mixed reality environments (MRE). An MRE comprises a virtual representation of the reality as well as physical objects augmented with virtual features. These objects are coupled with the virtual representation so that they can reflect its changes in real time. The proper design of an MRE and the user models that it implies are crucial for its success, but unfortunately, there are no guidelines for the design of these environments. In this paper we present a methodology for designing user models for MRE as well as for the augmentation of physical everyday objects. The user model describes users' knowledge in two levels of abstraction: objects manipulation (syntax) and its meaning assigned by a community of practice (semantics).

1 Introduction

User models could be defined as models that a system have about users, which reside inside the computational environment. An advantage of this approach is that a system can adapt itself to the current task or user, dynamically and with little effort or none required from the user [6]. This property is interesting mostly in complex

environments such as mixed reality environments (from now on MRE). MREs blend the real and the virtual [19] by keeping virtual representations of real things and introducing virtual features in the real world through complex objects. A complex object has a real concrete part coupled to various virtual representations (simulation, animation, symbolic) by means of grasp or image recognition [3].

Although research in MRE [15, 18, 22], tangible user interfaces (TUI) [9, 12], and multimodal interfaces [14], look promising, they cannot be considered by default beneficial. Bad designs lead to unnatural interfaces, hard to understand, requiring an extra cognitive effort from users [20], for manipulate them (syntax), and understanding the results of such manipulation (semantics). Unfortunately, most experiences reported in TUIs and MREs, are mainly ad hoc design strategies [14] instead of the general design frameworks of GUI interfaces research [16]. In addition, research in novel interfaces does not consider real contexts of use. As a result, there is not a clear understanding of users' needs, restrictions, knowledge and assumptions in relation with the interface. This situation avoids making a proper evaluation of interfaces impact on users and many times "hammers in search of nails" are created. For the case of user models in MREs, this situation is challenging because if the system decides to adapt itself according to a misconceived user model the resulting action could be performed in the physical world of a user and turn into a odd action.

How can we identify the most important aspects to consider when designing an MRE?. A first notion could be "transparent artifacts", it states that a well-designed artifact (such as a door) becomes transparent when it is used: it allows us to focus on the task at hand instead of on the artifact itself (e.g. a door allows us to focus on our plans such as getting into the kitchen instead of on the door itself) [1]. This concept is used in GUI interfaces design: it exploit users' knowledge about the world such as pointing, grabbing and moving objects [22]. Such knowledge can be understood as the perceived objects' properties in order to manipulate them or affordances (e.g. a file can be grasp through a hand icon) [8], and the expected results (e.g. erase a file).

But users' knowledge about artifacts goes beyond its physical manipulation; users assign meaning to objects based also on its use [1]. For instance, GUI designers facilitate the users' understanding of its actions, by exploiting common knowledge and organizing GUI elements into metaphors (e.g. providing a trashcan icon for files deletion). Particularly, *everyday objects* (EO) such as keys, doors, rooms, etc. have a meaning shared by a specific community. Users have expectations about them: a lawyer may expect to find his office door closed, while students may expect to find their room door opened. Just as we use basic knowledge such as pointing, grabbing, etc, and semantic knowledge such as metaphors in GUI interfaces, we can exploit EOs semantics for creating MREs with user models encapsulating this knowledge so that the system can adapt itself without disturbing the user.

In this paper, we present a methodology for guiding the design of a cognitive user model in order to enrich EOs comprised by a MRE. Our methodology has three main tasks: *syntax modeling*, which consists of characterizing EO manipulation (e.g. grabbing); *praxis modeling*, which consists of identifying the shared meaning assigned by a community to the EO; and *object augmentation*, which consists of determining the new virtual features of the object and its impact on the original object's syntax and semantics.

The resulting MRE comprehends a virtual representation of the real world, a set of complex objects immersed in the users' real world and a set of user cognitive models (one for each user). A *user model* encapsulates the user knowledge and assumptions about complex objects at two levels of complexity: its manipulation or syntax (grasping, moving, etc.), and the semantics associated with its *use* in a determined context [1]. As an example, we used our methodology for guiding the design of a MRE called "Collaborative Virtual-Real Environment" or CVRE [7, 10]. CVRE includes a virtual representation of the real facilities of our Department of Computer Science at University of Chile, a set of complex objects (the real part is coupled with the virtual by means of grasp recognition) and a set of user models implemented through software agents.

The rest of the paper is organized as follows: section 2 describes the conceptual background of our methodology. Section 3 shows the proposed cognitive user model for MREs. In section 4 we present the use of our methodology in a practical example. In section 5 we describe our CVRE. Finally, section 6 presents some conclusions.

2 Contextual knowledge: syntax and semantics

In the area of context-aware computing, user context is described as the conditions associated to the user's current location, such as: social aspects [5], physical properties [12] or related information [4]. More generally, *context* can be understood as "*the interrelated conditions in which an event, action, etc. takes place*"¹. In Artificial Intelligence (AI), context is used for *interpreting* the meaning of a sentence. For instance, if a friend asks us to "close the window", in a *cold, windy day*, we may understand that s/he refers to a physical window instead of a MS-Window. This way, context narrows down the proper interpretation of an expression [2]. In groupware, contextual information is provided to group members so they can *understand* how their actions fit into the group goals and choose the appropriate response among a set of possibilities [21]. In all these scenarios, context is used to determine the meaning of a situation, a sentence or an action, so that an appropriate response can be built.

In HCI, Bannon [1] proposes that objects should not be studied only as "things". Objects have no meaning in isolation: they are given meaning only through their incorporation into social praxis. This way, objects' meaning depends on the *context of use* of such object. We define this context as "the interrelated conditions in which an individual interact *purposely* with such object". Such conditions can be differentiated at least in two complexity levels: the manipulation or actions performed by users on the object (*syntax*) and the interpretation of its results or consequences (*semantics*).

According to Bannon, this semantics would depend on the community who uses the object. For instance, a regular family uses a frigidaire to store food, but in a hospital (another context of use), people can use frigidaire to store blood samples.

¹ Excerpt from Merriam Webster On Line at <http://www.m-w.com> .

In that way, communities determine different contexts of use and meaning for objects. The objects and their manipulation may be the same (syntax), but users' knowledge about the results of their manipulation depends on the context of use.

Others, like Brezillon [2] considers that contextual knowledge has two aspects: static knowledge, which remains constant throughout the interaction; and dynamic knowledge that changes throughout the interaction. Consider for example, an everyday object such as a pencil. The knowledge associated to its manipulation (e.g. how to hold a pencil) is mainly static and can be used in diverse contexts (e.g. grading students test, setting appointments in a PDA touch screen, etc.), however, when used it may serve to draw lines or to pick up a file (e.g. in the touch screen), the dynamic nature of context serves as a mean for supporting users' diverse goal or to allow users to assign unexpected uses to objects.

Finally, a fundamental guide for understanding the knowledge associated to the functioning of objects from a cognitive point of view, is provided by Norman [17]. He defines a series of concepts such as affordances (the perceived properties of a thing that determine how it could possibly be used), constraints (the perceived properties of a thing that prohibits some activities and encourages others), feedback (the perceived properties of a thing that permits sending information to users about what action has been done), etc.

Objects allow to share and divide work practice among people, mediating the people' work. This is particularly important in groupware where a well-designed shared object can help users to understand their work and choose a proper behavior, providing a better collaboration scenario. When people share a common physical space but interact in an asynchronous way objects become the elements through which people leave traces of their actions and intentions. In all these cases people' actions on objects are interpreted in the context of use that the particular worker's community shares and allows them to coordinate their actions.

2.1 Dimensions for Analysis and Design

Norman's concepts are useful for describing an object manipulation, but they are too general. With the aim of obtaining more specific guidelines, we followed Gutwin's and Greenberg's strategy [11]. They used five "type of questions" iteratively (what, who, when, how, where), for defining some dimensions of analysis and modeling of groupware context (e.g. identity, location, etc). After some iterations, refinement and discussion we found some useful dimensions for MREs. They are: Usage, Feedback, History, Intention, Consequence, Action, Dependence, Opportunity, Access, Roles, Reach and View. Each dimension must be defined in the two levels of abstraction discussed previously: syntactic (manipulation) and semantic (interpretation of manipulation by a community) level.

Usage: When referred to syntax, it describes the mechanism for manipulating an object. It could be obtained by answering a "How" question: e.g. How do you manipulate a key? (related to Norman's affordance concept, as well).

Feedback: When referred to syntax, it describes the mechanism to know when an action has been done. It could be obtained by answering “How” questions: e.g. How do you know the consequences of using a key? (Norman's feedback concept).

History: When referred to syntax, it provides information about past events concerning presence, location or action. It could be obtained by answering “Who” questions: e.g. Who used a key?, “Where”: e.g. Where is the person that used a key?, and “How”: e.g. How do you know if a key was used (Norman's constraints concept).

Intention: When referred to syntax, it describes the object properties that a user expects to change when interacting with the object. It could be obtained by answering “What” questions: e.g. What is a key used for? (Norman's affordance concept). When referred to semantics, intention describes the meaning associated to an object property change. It could be obtained by answering the question “What”: e.g. “What is the user intention when using a key?”.

Consequence: When referred to syntax, it gives information about the actions the user can predict when perform an action over an artifact. It could be obtained by answering “What” questions: e.g. What is the direct consequence of use a key?

Action: When referred to syntax, this aspect provides information about the state or process of doing something. It could be obtained by answering “What” questions: e.g. What do I do with a key?

Dependence: When referred to syntax, it describes the state of being determined, influenced, or controlled by something else. It could be obtained by answering “What” questions: e.g. What is the dependence of a key with people?

Opportunity: When referred to syntax, it represents favorable or advantageous circumstance or combination of circumstances of doing something. It could be obtained by answering “When” questions: e.g. When is a key used?

Access: When referred to syntax, it indicates the permissions of the people of use certain artifacts. It could be obtained by answering “Who” questions: e.g. Who can use a key?

Roles: When referred to syntax, it presents the characteristic and expected social behavior of an individual. It could be obtained by answering “Who” questions: e.g. Who should use a key?

Reach: When referred to syntax, it describes what is reached when an object is used. It could be obtained by answering “Where” questions: e.g. Where does a key allow to reach?

View: When referred to syntax, it describes what is viewed when an object is used. It could be obtained by answering “Where” questions: e.g. Where does a key allow to view?

3 Cognitive User Model for Designing Mixed Reality

In this section, we present a methodology for designing adaptive MREs. It comprehends five steps implemented in three phases: everyday objects syntax modeling, praxis modeling, and augmenting objects.

3.1 Phase 1: Everyday objects syntax modeling

Our aim is to design MREs that include everyday objects augmented with virtual features. By augment, we mean to manipulate a physical artifact so that it is publicly, and in most cases permanently, recognized to represent or denote something else. This kind of natural augmentation is an activity that human beings perform constantly. Our first step is to determine which objects will be considered as part of the environment. One of the risks when augmenting objects with new functionality is that we distort objects' syntax and semantics in a way that we lose useful properties or change the object so much that users may need extra cognitive effort to use it.

In order to avoid this, we model the object real syntax (manipulation) and semantics (interpretation) using the dimensions described in section 2.1 (numbered circles 1 and 2 in fig. 1). In this way, we can perform later a controlled distortion.

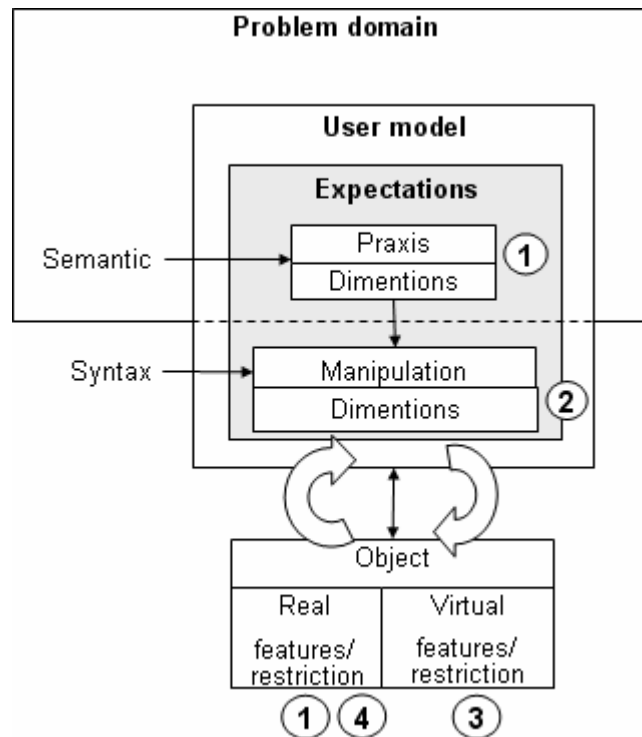


Fig. 1. An outline of the methodology followed for augmenting objects. A user cognitive model about an artifact comprising syntactic and semantic aspects is initially created. The model is used as a basis for performing a progressive and controlled distortion of the object.

3.2 Phase 2: Praxis modeling

Diverse communities will assign different meaning to the same object: for instance in a university, if an aula door is left half-opened, one may expect that students will get in as they arrive; however, if the door belongs to a lawyer office, one may interpret it as a potential security hazard. Furthermore, within a community we can find different types of users or roles, such as teachers, secretaries, students, etc., whose praxis may be slightly different for each one.

A first step is to identify the set of users or roles in a community. For each of them, we must characterize their knowledge about objects usage. Again we use our dimensions for finding objects semantics (numbered circle 3 in fig. 1). In figure 1 we can observe that syntax and semantics are separated with a dotted line. Indeed, we can see that semantics are included in a box labeled “problem domain”. This is because syntax or manipulation can be the same across diverse domains of use, for instance in the door example, the artifact (door) can be manipulated with the same set of actions in a school or in a lawyer office (assume that the object manipulated is the same). The opposite applies for shared meaning of object manipulation; diverse communities will assign another meaning to them, depending of their use context. For instance lawyers and teachers could assign different meanings for a structurally similar half-opened door.

Semantics may be expressed as shared policies: users of a certain type agree on a specific interpretation of objects usage and needs. For instance, secretaries may agree that the doors of their offices must remain opened during their work-time. This knowledge corresponds to the assumptions that the system takes into consideration when adapting its behavior.

3.3 Phase 3: Augmenting objects

The goal of the phases previously presented is to identify the objects in order to augment, their physical properties, weaken their restrictions and manipulation constraints, as well as to change the expectations hold by each type of user in relation with each object. In this stage, we define the desired objects’ virtual features (numbered circle 4 in fig. 1).

These features should be consistent with the syntax and semantics defined in the previous steps. A designer may choose to change some of them, but s/he will know in advance that users may need to learn how to use these new features. As well, a designer may choose to modify an object (numbered circle 5 in fig. 1). For instance, s/he could add leds, speakers, motors, etc. Again s/he should consider the impact of his/her choice on syntax and semantics. If the object is modified, then its physical constrains and manipulation could change. Furthermore, users may decide to change their shared policies in order to take advantage of objects new possibilities. In this case the cycle must be followed again (cyclic arrows in fig. 1).

In the next section we apply our methodology for creating a MRE. The environment comprises the physical workplace of the Computer Science Department at the University of Chile that has a counterpart in the virtual world in the form of a

Web based Collaborative Virtual Environment. Additionally, physical elements such as doors, keys and rooms have been augmented following the methodology.

4 Designing a CVRE

In a previous work we have designed a Collaborative Virtual Environment (CVE) [10], which is a virtual space where people can collaborate. The CVE visually mimics the real world in order that people can use it in a natural way. Now we will extend our previous work and transform it into a MRE (CVRE) comprising augmented everyday objects.

4.1 Everyday objects syntax modeling.

The everyday objects considered as part of our CVRE are rooms [10], doors, door-locks and keys. Rooms are virtual representation of the real workplaces in the Computer Science Department: X teachers' offices, Y students' rooms, etc. Rooms are assigned to one person (owner). Access to rooms is controlled by one wood door without glasses and one door-lock. Door keys are assigned to room owners. The administrator keeps also a copy of each key. A properties summary is shown in Table 1.

Table 1. Our CVE contains rooms, doors with locks and keys. Their real features are described in this table

| Room | Door | Lock | Key |
|---|---|---|--|
| - workspace: academic office, secretary office, aula room, meeting room, etc. | - it allows to the enter into a room. | - it allows to close a door. | - it allows to leave closed a lock. |
| - it can have glass walls (transparent) or not. | - it can be open, semi-open or closed. | - it is used with a key- equal locks are allowed. | - we can obtain a key copy. |
| - assigned to 1 or more people. | - it can have glass walls (transparent) or not. | | - is transferable. |
| - it has a door. | - it has a lock. | | - it is possible to be taken to all parts. |

4.2 Praxis modeling

CVRE users are: Professors, Administrative personnel, Research assistants, Teaching assistants, Students and other academic personnel. In table 2 we present the praxis modeling for Professors in relation with an office door.

Table 2. Praxis model of a door office. The table shows the detailed analysis of the shared meaning assigned by teachers to their doors. The user model will contain some of these dimensions as rules used to adapt CVE to users preferences. Notice that at the semantic level, teachers' offices or rooms are workspaces.

| Dimension | Question | Answer (Door) |
|------------------|---|---|
| History | How do you know if a door was used? | If either the door status, the workspace content or the workspace status has changed since last visit. |
| Intention | What is the objective to open a door? | To enter into a closed workspace. To change the visibility of a workspace. To allow that some person leaves the workspace. To allow people to inspect workspace content. |
| | What is the objective to close a door? | To close an opened workspace. To change the visibility of a workspace. To avoid others to leaves the workspace. To avoid others to inspect the contents of a workspace. |
| Consequence | What is the direct consequence of keep the door opened? | Passers-by can contact people inside the workspace. Workspace contents are visible for everybody. A person is allowed to leave the office. |
| | What is the direct consequence of keep the door closed? | Hide the content of the workspace. Users must knock the door, for knowing if anybody is inside. We do not know who is outside room. |
| Action | What do you do with a door? | Enter / leave a workspace. Allow / deny the visibility of the workspace contents. |
| Dependence | What is the dependency with people? | Regularly, people who open a door, have sufficient permissions to enter into the workspace. |
| Opportunity | When is a door used? | When a user needs to enter/leave a workspace. When a user needs that other people enter/leave workspace. |
| Access | Who can use a door? | The person who are next to a door and need to open/close it. |
| Roles | Who should use a door? | The person who is allowed to open/close it. |

In table 3 we present an analysis of the syntax modeling of a key. Note the differences with table 2: the answers are described in terms of physical properties and not in terms of the changes that we performed.

Table 3. Syntax model for a door office key. The table shows the detailed analysis for understanding key manipulation. Possible key status will be also contained in the user model.

| Dimension | Question | Answer (Key) |
|------------------|--|--|
| Usage | How do you manipulate a key? | Putting the key in the door lock. |
| Feedback | How do you know that it is the correct key | Because the key fits the door lock. If it does not, it is not the correct key. |
| History | How do you know if a key was used? | I cannot be sure. |
| Intention | What is a key used for? | Opening/closing door locks. |
| Consequence | What is the direct consequence of using a key? | Door locks change their state to open/close. |
| Action | What do you do with a key? | Putting the key in the door lock, turning it, and removing the key. |
| Dependence | What is the dependency with peoples? | Only the carrier of the key can use it. |
| | What is the dependency with door-locks? | Only the key that fits the door lock can be used. |
| Opportunity | When is a key used? | When I have the key that fits a door lock and I want to open/close the door of a room. |
| Access | Who can use a key? | The people that have a key. |
| Roles | Who should use a key? | People that have a key and need to open the a door lock. |
| View | Where does a key allow to view? | The contents of the room associated with a door related with a key (transitive). |
| Reach | Where does a key allow to reach? | To the room associated with the key. |
| History | Where is the person that used the key? | Possibly in the room, but I cannot be sure. |

4.3 Augmenting objects

Table 4 describes the virtual features that we have chosen for augmenting some objects. The most important feature is the creation of a desktop-lock, which is basically a door-lock that allows us to open and close virtual shared workplaces by

using an augmented key. Unlike the real world, desktop-locks are not associated with just one key, but with many keys as long as they belong to a valid virtual workplace.

Table 4. . Virtual features of rooms, doors with locks and keys, chosen according properties discovered in tables 2 and 3

| Room | Door | Lock | Key |
|--------------------|--|---|---|
| virtual workspaces | new intermediate state: semi-opened door. (which define a semi-accessible space) | desktop lock: allows to open and close virtual shared workplaces. We can use the same lock for various rooms (only change the keys) | activity log (register past event about use of a key) |

5 Implementing the CVRE

We have used phidgets for augmenting everyday objects (e.g. the key-lock pair). Phidgets (physical widgets) are specialized devices developed at the University of Calgary that leverages the complexities of developing physical interfaces [9]. The philosophy behind Phidgets is to resemble the GUI widgets; they are GUI elements that encapsulate interface interaction and make GUIs easy to develop as they may be arranged for composing an application interface. Phidgets encapsulates minimal functionality for rapid prototyping of physical interfaces. The elements we have used for implementing our CVRE are: a Phidget Interface Kit (the main interface where all the sensors are connected to); a rotation sensor (which allows us to control the twists of the key in the lock); three LEDs (allowing us to give feedback about the accessibility state provided by the key); a RFID (Radio frequency identification, a small object attached to a key that allows us to read information associated with each key in order to identify them); a movement sensor (that allows us to know if the user leaves his keys in the lock at the time of leaving his job). Fig. 2a depicts the augmented interface built.

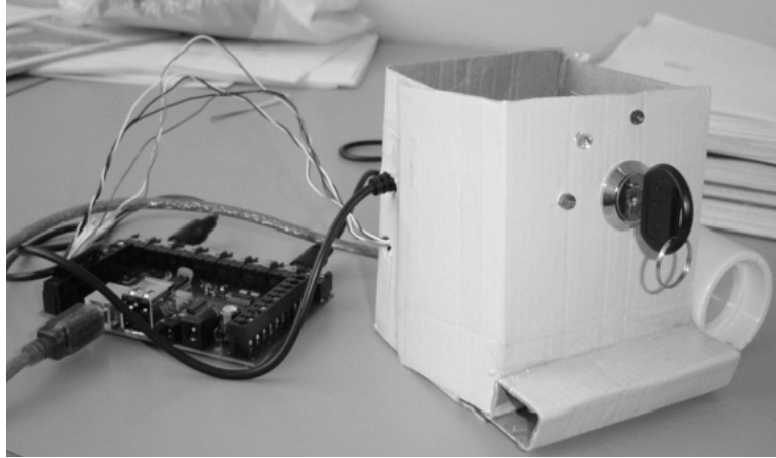


Fig. 2. Key interface developed using phidgets

Three LEDs (a red, green and yellow ones), indicate the accessibility provided for a particular room. The green color represents a totally opened lock (a fully accessible space); the red color represents a closed lock (a totally inaccessible space), and the yellow color represents the intermediate state previously described (a semi-accessible space). A tag has been attached to each key, so that it makes possible to the RFID reader to identify and differentiate the used keys. For this reason, it is necessary that the keys associated with different virtual rooms be physically equal (in order to be accepted by the regular lock). The tags attached to them differentiate keys.

A movement sensor is located next to the lock, facing the user (the white circle on fig.2a), in order to detect his/her presence or absence.

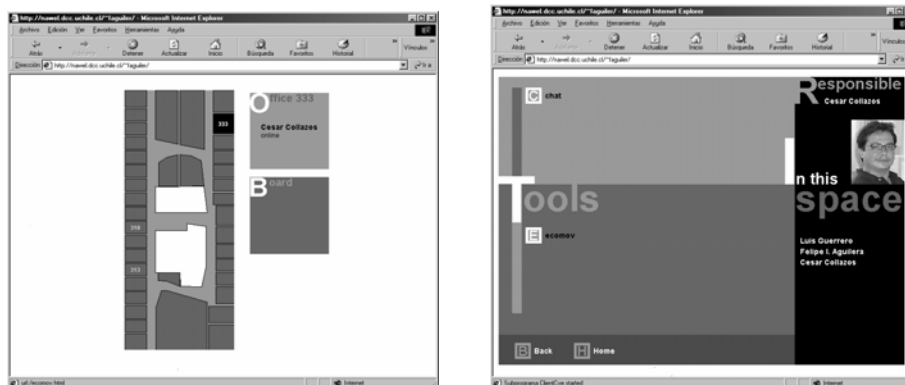


Fig. 3. CVE main interface corresponds with the floor plan of DCC. Keys are used to allow a user to access his or her personal space

CVE virtual rooms. Our CVE [7] maps the real physical design of our Computer Science Department and allow CVE users to be aware of such structure through a floor map (Fig. 2b and 2c). Users can access into their own personal space (a room) or get into the virtual workspace of a colleague. Again we follow our guidelines and exploit the knowledge that users have about the structure and division of work that actually occurs at our Computer Science Department.

In our CVE, users employ the desktop-key to open and close virtual spaces. Doors allow users to control their privacy and the degree of availability that a user wants that by-passers perceive about him. Our key-lock interface allows a user to control the state that the user wishes others to see in the virtual interface. For it, each position of the key corresponds to a state: totally closed (red) to indicate that s/he is offline, totally opened (green) to indicate that s/he is available, and an intermediate state (yellow) to indicate that one is busy or temporarily absent. This alternative was used to control the state of a user of a session of instant message.

6 Conclusions and Further Work

The presented work stresses out the importance of the design of physical interfaces that take into account the previous users' knowledge about the objects' context of use. Such context comprises both syntactic and semantic categories of analysis. There has been a large tradition of discontinuity between the rich interactions with objects in our physical world, and the impoverished interactions with electronic material. The linkage between these two worlds has been difficult and expensive. But we believe that rather than force users to adopt radical forms of interaction, we may exploit instead the achieved knowledge about how the world works. By augmenting pre-existing tools within the framework of a work praxis or context of use, we may bring computing to the world instead of the other way around. By designing interfaces that augment objects' features, but keeping a coherent semantics with the original version and its usage in the real world, we believe that a more natural design of these interfaces could be achieved.

On the other hand, computing nowadays provides enormous potential for novel, unexpected, rich and useful interaction. We do not argue against such approach, however, we believe that the presented work may serve as a reference for identifying the successful design choices and its rationale as well as to suggest needs and promising research areas.

In this paper we have presented a framework for analyzing the context of use of a physical everyday object in order to identify its associated semantics, augmenting it (i.e. by adding intermediate states) and use those semantics for new tasks (i.e. by allowing to handle multiple context of work or providing availability awareness). We have used phidgets as a medium for augmenting and manipulating everyday objects easily (i.e. a key as an authentication mechanism).

Other issues we want to explore as future work, are the possibility of dynamically selecting devices' relationships. For instance, a key can handle more than one application, but the lock is physically the same for each door. Hence, we want to explore the possibility to enrich the key so that it can provide multi sensory

information (i.e. change its color), so that a key may represent indeed a set of keys. May the users perform a context switch because of the color? The research question for further work aims to identify more context attributes. For instance we may argue that users' intention when manipulating an object actually triggers the users' appropriate context for interpreting, let us say, an action. Otherwise, physical (sensory) arrangement of physical environment may trigger users' appropriate interpretation context.

Finally in order to answer the questions raised in the discussion as well as to learn more on the effects of this approach on users, we need to design appropriate tests and evaluate the objects' usage in various context of use.

References

1. Bannon L., Bødker S.: Beyond the interface: Encountering artifacts in use. In: J. Carroll (ed.): *Designing Interaction: Psychology at the Human-Computer Interface*. Cambridge: Cambridge University Press 1991, pp. 227-253.
2. Brézillon, P. and Abu-Hakima, S.: Using knowledge in its context: Report on the IJCAI-93 Workshop. *The AI Magazine*, 1995, 16(1), pp. 87-91.
3. Bruns, F. W.: Complex Objects and Anthropocentric Systems Design. In: *Advances in Networked Enterprises* (L. M. Camarinha-Matos, H. Afsarmanesh, H.-H. Erbe (Eds.)), Boston, 2000, pp.249-258.
4. Cheverst, K., Davies, N., Mitchell, K., Friday, A., Efstratiou, C.: Developing a context-aware electronic tourist guide: some issues and experiences. *CHI'00*, pp. 17-24.
5. Dey, A. K., Salber, D., Abowd, G. D.: A Conceptual Framework and a Toolkit for Supporting the Rapid Prototyping of Context-Aware Applications. In Moran, T.P. and Dourish, P. (Eds.) *Context-Aware Computing: A Special Triple Issue of Human-Computer Interaction*. Lawrence-Erlbaum, March 2002.
6. Fischer G.: User Modeling in Human-Computer Interaction, In *User Modeling and User-Adapted Interaction*, *UMUAI*, 11(1), 2001, pp. 65-86.
7. Frecon, E., and Nou A.: Building distributed virtual environments to support collaborative work. *ACM Symposium on Virtual Reality Software and Technology, VRST'98*, pp.105-113, Taipei, Taiwan, Nov. 1998.
8. Gibson, J. J.: The theory of affordances. In R. E. Shaw and J. Bransford (Eds.), *Perceiving, Acting, and Knowing*. Hillsdale, NJ: Lawrence Erlbaum Associates, 1977.
9. Greenberg, S., and Fitchett, C.: Phidgets: Easy development of physical interfaces through physical widgets. *Proceedings of the ACM UIST 2001 Symposium on User Interface Software and Technology*, November, Orlando, Florida, ACM Press, 2001.
10. Guerrero, L. A., Collazos, C. A., Pino J. A., Ochoa S. F., and Aguilera, F.: Designing Virtual Environments to Support Collaborative Work in Real Spaces, *Journal of Web Engineering*, 2(4), October, 2004, pp.282-294.
11. Gutwin, C. and Greenberg, S.: A Descriptive Framework of Workspace Awareness for Real-Time Groupware. *Computer Supported Cooperative Work*, 11(3-4), 411-446, Special Issue on Awareness in CSCW, Kluwer Academic Press, 2002.
12. Ishii, H., and Ulmer, B., Tangible bits: Towards seamless interfaces between people, bits and atoms. *Proceedings of the ACM CHI'97*, pp. 234-241, 1997.
13. Leech, G.: *Semantics: The Study of Meaning*. Harmondsworth, UK: Penguin, (1981).

14. MacLean, K. E. and Roderick J. B.: Smart Tangible Displays in the Everyday World: a Haptic Door Knob. Proceedings of the IEEE/ASME International Conference on Advanced Intelligent Mechatronics, AIM'99, September 1999, Atlanta, USA.
15. McGee, D.R., Cohen, P.: Creating Tangible Interfaces by Augmenting Physical Objects with Multimodal Language. Intelligent User Interfaces, 2001, pp.113-119.
16. Nielsen, J.: Usability Engineering. The Computer Science and Engineering Handbook 1997, pp. 1440-1460.
17. Norman, D. A.: The Design of Everyday Things, London/New York: MIT Press, 2000.
18. Olwal, A. and Feiner S.: The Flexible Pointer: An Interaction Technique for Augmented and Virtual Reality, UIST 2003, Vancouver, BC, November 2003, pp. 81-82.
19. Ohta, Y., Tamura, H.: Mixed Reality—Merging Real and Virtual Worlds. Tokyo, 1999.
20. Shneiderman, B.: Direct Manipulation for Comprehensible, Predictable and Controllable User Interfaces. Proceedings of International User Interfaces, 1997, pp. 33-39.
21. Sohlenkamp, M.: Supporting group awareness in Multi-User Environments through Perceptualization. GMD Research Series, No.6 Zugl.: Paderborn, Univ. Diss, 1999.
22. Want, R., Fishkin, K.P., Gujar, A., and Harrison, B.L.: Bridging physical and virtual worlds with electronic tags. Proceedings of the Conference on Human Factors in Computing Systems, 1999, ACM Press, pp.370-377.