An Analytical Frame for Describing Multi-* Interactive Systems

Augusto Celentano, Emmanuel Dubois

Official URL
https://doi.org/10.1145/3351995.3352040

An Analytical Frame for Describing Multi-* Interactive Systems

Augusto Celentano
DAIS, Università Ca' Foscari Venezia, Italy
auce@unive.it

Emmanuel Dubois
IRIT, Université Toulouse III, France
emmanuel.dubois@irit.fr

ABSTRACT

In this paper we propose a 3D space describing and structuring the facets of user experience in systems characterized by multiple devices, locations, media types and interaction modalities, that we call *multi-* systems*. The space is structured around three dimen-sions: *Actions*, *Information* and *Environment*. We discuss the ra-tionale for such choice and show how some fundamental princi-ples of Software Engineering can support the design of HCI com-plex applications modeled according to this user experience space.

KEYWORDS

Interaction design, Distributed interaction, Multi-device, Software engineering

https: //doi.org/10.1145/3351995.3352040

1 Introduction

The scenarios in which complex interactive applications are developed and used have notably changed in the last decade. The growth of digital applications in a large spectrum of domains, the spread of embedded systems, ubiquitous networks of sensors, and ambient augmentation have produced an increase of information available both for direct processing within dedicated applications and indirectly, as a context to users for personalized experiences. The rapid evolution of mobile systems, perceptual interfaces, tangible systems, the Internet of things, have moved the user interaction possibilities far beyond the single machine operations. Multi-device applications and environments executing distributed applications, accessing multidimensional and multimedia information in multiple locations over variable spans of time are today common [13, 17, 20, 25].

We introduce the term *multi-** to denote such interactive scenarios. In such systems the user changes task and location and uses several devices according to the varying needs of a complex procedure. Interaction develops along dynamic sequences that depend on the technology used and the environmental context. The information accessed changes in content, detail, representation and deployment on devices. While the goals and applications domains of such systems are wide and differentiated, to delimit the area of discussion to a reasonable size we focus on information access in such scenarios.

A set of relevant issues comes immediately to our attention, in the form of questions concerning both the analysis of the domain and the design of applications and interactive experiences. First, how to analyze such situations, i.e., how to identify the components of such complex user experiences, and their relations and connections? And then, since the design involves complex issues due to the large variance of the technical solutions and the wide background needed to compare and evaluate them, how can we match the design goals with the multi-* environment?

The ultimate aim of this work is to contribute to the identification of the most appropriate interaction solutions supporting such scenario: (1) by taking into account the specificities and constraints of a multi-* system perspective, as they are more numerous and variable due to the multiple nature of the environment; (2) by taking advantage from experience, models, tools and practices in other fields of complex systems design and development.

The contribution of this paper is the proposal of an analytical frame of reference for reasoning about multi-* interactive systems, based on a 3-dimensional space <*Actions*, *Information*, *Environment*>. The dimensions characterize, respectively: (1) the user activity and its structure, (2) the information accessed and processed, and the way it is organized and viewed, (3) the (physical) environment in which the user moves and acts. We call such space the *user experience space*. Points and regions in it describe different aspects concerning the user experience at different levels of detail. Links connecting points and regions describe the dynamics of the user experience during the execution of an activity.

2 The User Experience Space

The choice of the three dimensions <*Actions*, *Information*, *Environment*> is not casual. As established in the Norman's Action theory [18, 19] when using an interactive systems, any user is looking to reach a goal and will necessary have to form an intention and perform actions to reach this goal: the granularity considered to describe these actions leads to the identification of different potential points of view that are addressed through the *Action* dimension. The second dimension, *Information*, is tightly coupled to the focus of our work, i.e., accessing a rich set of information: given the huge potential variety (origin, nature, type, etc.) of information, better understanding which part of the set the user is dealing with is covered by this dimension. Finally, as the collected data and the access to it is distributed on several devices possibly located in various physical places, the *Environment* dimension identifies the different parts of the user's environment that play a role in the interactive experience.

The three dimensions are orthogonal in principle, but specific applications and domains can show dependencies and constraints among an action, the location in which it is executed and the way data is accessed. The user experience space is able to capture such dependencies and constraints which appear as non-homogeneous distributions of points and regions describing the development of a specific activity.

Each of the three dimensions is structured into subcategories, recalling principles and theories described in the literature and applied to the analysis and design of complex human systems. We present these refinements in the following subsections.

2.1 The Actions Dimension

As an extension of the concept of action according to Norman, we refer to the *Activity Theory* (AT) as a more general framework that describes the human work in terms of mediated interaction between human beings and the world [1, 16]. According to the AT the human *activities* are structured in *tasks*, which are composed of *actions*. In the context of this work we use the term *operation* to refer specifically to actions on interactive devices. A further level of detail is represented by the *gestures* that a user does for executing an operation, but we shall not enter into such a level of detail which is relevant only for the design and analysis of low level operations concerning interface implementation.

Activities, tasks and operations have different relations with the space, which is distributed (for activities), local, or device-bound (for operations). They also involve several levels of interaction: a high level (corresponding to the activity) supports a complex interactive experience and is driven by the user's motivations; an intermediate level (corresponding to the execution of a task) supports a goal-directed interaction; a low level (corresponding to the operation) is characterized by properties like speed, efficiency, ergonomics, etc. Such hierarchy is recognized also by other models of the human work, such as the Hierarchical Task Analysis [23], and is therefore a suitable dimension to describe a space of complex user experiences.

2.2 The Information Dimension

The Information dimension describes the views through which a user can access and process information. In general, during a multi-* interactive experience the user accesses several information contents, at different levels of detail, with different representations, for different goals. Information is presented and processed in several steps, whose correspondence with the user activity, tasks and operations is established by the application design at some extent, and by the amount of user freedom in interaction. The technology and the environment add further elements of variability. The relations between the information content and its representation, the refinements in levels of detail, the association with the interaction functions become complex. The design choices are often left to the interaction designer's skill, but in a complex multi-* paradigm a more solid background is required.

In a previous paper [3] we discussed a design space for rich interactive multimedia content, organized into five layers based on the identification, definition and characterization of five categories of information, targeted to different knowledge goals and needs. The outermost category is the *Universe*, which defines a knowledge domain and encompasses the whole information processed by some interactive application. The innermost category contains the elementary data *Items*, the smallest elements of information accessible. Three more layers are the *Scenario*, the *Region of interest* and the *Compound* (an aggregation of elementary data items). We draw from that work such information layering, simplified and re-organized into the three layers *Scenario*, *Region of Interest*, and *Data*. We take apart the *Universe* and remove the distinction between *Compound* (a structured set of elementary data) and *Item* (an elementary data instance), which is relevant only for structured data and aggregate/element operations.

In our experience space the *Scenario* describes the widest view on information relevant for a user with respect to a specific goal, and subject to a uniform perceptual representation. In terms of an often adopted overview-detail hierarchy the *Scenario* is at the overview level. It can be examined, modified but the details of data are not accessible at this level.

The *Region of interest* (*ROI*) defines a view on the *Scenario* containing information subject to a specific exploration in the context of some activity or task. The *ROI* can be accessed by appropriate interaction techniques, possibly on devices different from those used for accessing the *Scenario*. Indeed, different *ROIs* represent different sets of information associated to different phases of the user work, or to different user's motivations and goals. A change of ROI can also correspond to a change in the information representation/interaction, or in the deployment on different devices.

The *Data* layer contains and exposes to the user the detailed information relevant for the user, presented and accessed according to visualization and interaction modalities appropriate for the information type.

The layered structure and the related terminology were conceived in [3] with reference to a spatial representation of a data domain, e.g., a map. It is, however, appropriate for any structured information domain, as it clearly appears if we replace the words

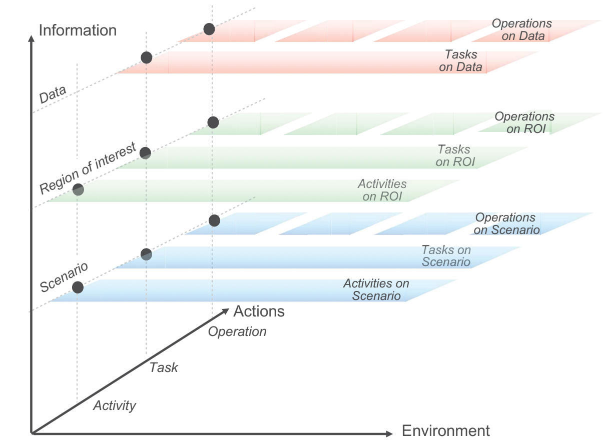

Figure 1. The user experience space

Figure 2. The use-case setup (from [4])

Scenario, *Region of interest* and *Data* with, e.g., *Database*, *View* and *Record*.

2.3 The Environment Dimension

The role and importance of the space in human activities has been recognized by many authors. According to [15], "space is fundamental to perception and cognition because it provides a common ground for our senses and actions". The association between the structure of the physical space and the interactive activities done in it has been qualified in [5, 21] through the concept of *Interaction Locus*, an association of information and environmental properties. The concept supports the structuring of an interactive environment in a set of places (*loci*) each devoted to a specific and recognizable user activity.

In our experience space we refer to an *interaction environment* structured in a set of *locations* (not necessarily hierarchically layered), in which tasks are executed. We consider a task to be executed on a specific device or set of devices in a certain physical location or, in case of mobile devices, through a sequence of operations with the same device, possibly in continuously changing locations. Physical locations are constrained by the relations between the physical ambient and the application functions (e.g., a security area where specific procedures must be followed) and by the availability of suitable devices and resources (e.g., a desktop equipment). "Floating" locations can be dynamically instantiated and changed during the execution (e.g., mobile devices can be seamlessly used in different areas). A border situation that we do not face in this work is the ambient computing in which the perception of the locations fades into a continuum space surrounding the user.

The lack of a spatial hierarchy is typical of distributed environments where the association between actions and locations depends on the deployment of the resources rather than on predefined structural properties of an activity, and distinguishes this dimension from the other two. It's possible to conceive hierarchical interaction spaces where locations are grouped to support sets of connected activities, tasks or actions. In our view such hierarchy does not add interesting properties to the ambient, and we shall not elaborate on this.

Given these three dimensions, their composition in a user experience space is illustrated in Figure 1: the *Information* axis holds increasing levels of information detail from the *Scenario* to the elementary *Data* structures. On the *Actions* axis increasing levels of detail from the user activity down to the operations performed on a specific device are placed. On the *Environment* axis an unordered, non-hierarchical collection of locations is placed. The shaded regions in the figure represent the actions and information layering. In principle, all the combinations of actions and information layers are possible, but in practice some of them are not plausible. In particular, given the complexity and size of the information space, it is unlikely that a specific elementary data is perceivable and is manipulated at the activity level: data selection and manipulation are typically done in the context of tasks and operations. In the *Environment* dimension the large layers represent a multiplicity of locations for activities and complex tasks, while the small layers represent single locations at the operation level.

3 Use Of The Experience Space

To give concreteness to the experience space use we present a use-case based on the *neOCampus* project developed at University of Toulouse (https://neocampus.univ-tlse3.fr). *neOCampus* is a multidisciplinary approach aiming at the definition of a smart, innovative and sustainable university campus. The project is based on a huge information system recording and storing instant and historical values about gas, electricity and water consumption, together with equipment technical data. Consumption data is collected at different levels, from buildings down to individual devices. A control room provided with a variety of devices (computers, mobile devices, large screens, interactive tables) hosts the control applications. For space reasons we cannot describe here the project details. We recommend the reader to consult a refined presentation of the project in [4].

We address a simplified version of the experiment described in [4]. Figure 2 shows the experiment setup, which is a lab replica of a part of the control room installation. The PC on the left is the console and is not used in this use-case description. The user activity consists in the examination of the energy consumption for

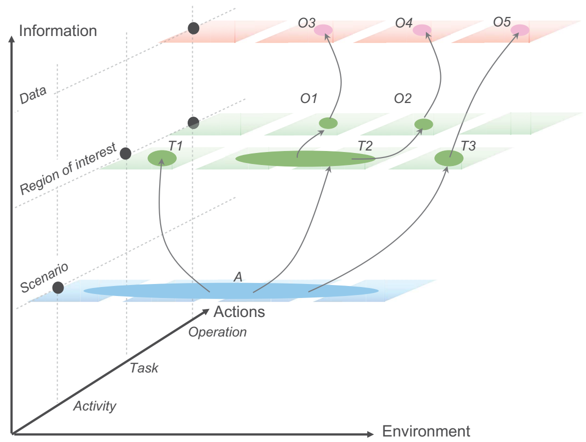

Figure 3. An example of activity decomposition

selected buildings and rooms according to various representations, using several devices: a multimedia table (in the lower part of Figure 2) showing a map of the campus on which buildings and rooms can be selected; a large screen (top of Figure 2) showing energy consumption values; a workstation (on the left) for executing a simulation of the energy management systems; a mobile device (placed on the multimedia table) for controlling the data shown.

Figure 3 shows the main components and phases of the user experience. The *Activity* spans several devices distributed in several locations of the control room, and refers to the *Scenario* made of the campus map with associated cumulative energy consumption values on the *Information* dimension. The *Activity* ("A" in Figure 3) is split in three tasks: map navigation and building or room selection ("T1"); energy related data selection and examination ("T2"); modification of energy management parameters ("T3").

Task T1 moves (in several steps through operations not shown in Figure 3) from the *Scenario* to the *ROI* level by defining which building/room is selected. Task T2 moves from the *ROI* level to the *Data* level through operations that select the data to be shown (power, gas and water consumption, instant or cumulative values, historical series, etc.) and their representation (e.g., numerical vs graphical). Operations may apply to cumulative data, as represented in Figure 3 at the *ROI* level by "O1" and "O2": for example, one floor might be represented in red if the consumption is too high. Operations can also apply to individual values and types of representation, as shown in Figure 3 at the *Data* level by "O3" and "O4"; for example, the detailed amount of KWh consumed may be displayed for each individual device of a room.

Task T3 involves the execution of a simulation program that changes the energy management parameters, allowing the operator to examine how they affect the energy consumption. This is accomplished through a set of operations at *Data* level (collectively denoted by "O5" in Figure 3). Each set of operation, e.g., open a property bar, select an icon, enter a threshold, etc., has an impact on different dimensions of the simulation, hence contributes to the realization of separate user sub-tasks.

The operations that compose each task use different devices, hence are represented in Figure 3 in different locations: the multimedia table for the operations of Task 1, the mobile system and the large screen for Task 2, and the simulation workstation for Task 3. The *Activity* execution spans all the locations.

In Figure 3, albeit severely simplified for readability reasons, the flow of execution, the change of information level examined and the change of devices are clearly visible, describing in a synthetic way how the user experience evolves.

4 Matching Software Engineering Concepts To The Experience Space

This analytical frame is a first brick towards the building of a theoretical support for reasoning about the growth of complexity in interfaces and interaction environments. However, ensuring the potential and validity of such a frame is not an easy task. As HCI is addressing more and more complex systems, we chose to rely on knowledge and principles well established in the domain of complex systems to reinforce the relevance and interest of our approach. In particular we have examined the Software Engineering (SE) domain as a source of inspiration and we discuss in the following how its principles match the structure of the user experience space introduced above, hence ensuring the ability of our approach to tackle the analysis and design of complex systems.

The literature reports several studies about the integration of SE principles and HCI, examined from different points of view: the integration of usability studies into SE [8, 11, 22]; the comparison of the requirements of the two domains and the active role of the user in modifying the system behavior even in unanticipated ways [2, 7]; the interdisciplinary approach needed to join human related concerns to the technical design issues [6, 9].

Individual design principles, good practices and cross-relations between SE and the HCI domains have been discussed at some extent [24, see also the above cited references]. There is, however, no systematic discussion of the specific issues posed by the multi-* interactive systems, nor evidence of an effort to integrate them into a unique, comprehensive view of multi-* systems design at several levels of detail.

We aim to fill this gap: we seek to establish here that our analytical frame conforms to/reproduces/adopts the main principles forming the core of SE and contributing to support the design of complex systems from a technical point of view. Among the main principles of Software engineering, *top-down design*, *design in-the-large vs design-in-the-small* and *modularity* have a primary role, contributing to ensure that a system can evolve and be maintained in a sustainable way. We claim that they can fit our experience space and be applied to multi-* HCI design.

4.1 Top-down Design

Interaction design should be layered in a sort of top-down fashion, starting from the design of the overall interaction environment concerning the different phases of the application and their interconnections, down to the choice of the relevant properties of the devices, their physical properties and the operations to be executed on them. Indeed, top-down design is the result of the systematic application of refinements to structured information, functions and interaction. That means to define a hierarchy of data specifications, access functions and interaction techniques such that at each level of the hierarchy they can be combined and evolve almost independently.

Our user experience space ensures such an approach by: (1) refining the activity into tasks and operations according to the Activity Theory, and (2) refining the information space into a layered organization of categories related to the task-operation hierarchy.

4.2 Design In-the-large And In-the-small

In software design the intra-module and inter-module programming are distinct activities, each with own methodologies, languages and tools [7]. Interaction design should address with different perspectives and models the conception and organization of the phases of a distributed procedure wrt the details of a single interactive interface. Some authors have introduced the term *interaction in the large* to denote long lasting interactive activities that benefit from the integration of different HCI models [12, 14], but they have not discussed the issues of the different design goals at such level wrt the interaction on a local scale.

A split between local and distributed interaction in distributed systems has been proposed by the authors of this paper in [4] through the concepts of interaction *in-the-large* vs *in-the-small* to describe the user's actions at different levels of granularity. The interaction *in-the-large* layer defines the sequence of tasks a user does moving from one phase to the next of a complex activity involving multiple devices and locations. The interaction *in-the-small* layer defines the sequence of operations done by a user on local devices in a short and continuous span of time, with a limited and well defined goal.

This duality is supported in our approach through the identification of: (1) the different places in which a set of interactive resources is available, and (2) their relations to the other dimensions, actions and, mainly, information. In particular, information layering in *Scenario*, *Region of interest* and *Data* contributes to delimit the borders in which and across which local and global actions are executed.

4.3 Modularity

In SE modularity comes naturally from the layering of different levels of refinement in a top-down design approach and from information structure design. In the HCI context the layering of actions and information of a multi-* system should result in the identification of *interaction modules* which, as software modules, encapsulate the detail of the interaction for a specific phase of execution. The difference is that an interaction module transforms an initial state of a part of the system into a final state according to the variable user behavior, not only according to an autonomous and predefined algorithm. In HCI the design of interaction is still largely the result of a creative process. Hence how modularization is designed is not dominated by the functional specifications of the application but involves also a number of non-functional requirements. These are defined, among other factors, by the foreseen interaction model, the device technology, the environment, the user profile, etc.

Like traditional software modules, an interaction module should include a description of the internal behavior (the implementation of interaction model of a task or activity), as well as an interface with other modules and with the environment (location, device properties, etc.) to assure not only a correct exchange of parameters for a correct execution but also the necessary continuity and coherence in the interaction across modules. This concepts and requirements needs to be further refined but are already compatible with our approach.

5 Conclusion

Facing the growing complexity of interactive systems, in particular those devoted to data exploration, we have introduced the concept of *multi-* interactive system*. In order to help reasoning about such systems, we have proposed a 3-dimensional space <*Actions*, *Information*, *Environment*> and refined its structure on the basis of well-established principles taken from the design of complex systems. Extending these principles to the context of multi-* interactive system and linking them to our user experience space confer to it an initial proof utility for the design of such complex system.

Future work will be targeted to a refinement of the *interaction module* concept, supporting the design of a multi-* interactive system as a set of interconnected and coordinated modules covering both the interaction and the logic functions of a set of tasks. Such an approach is similar to the concept of GUI widgets, modules defined at a software level supporting short-term interactions; interactive modules will play the same role but at the experience level.

REFERENCES

[1] O. Bertelsen and S. Bodker. Activity theory. In J. Carroll (ed) *HCI Models, Theories, and Frameworks: Toward a Multidisciplinary Science*, Morgan Kaufmann, 291–324 (2003).

[2] J. Brown. HCI and requirements engineering: Exploring human-computer interaction and software engineering methodologies for the creation of interactive software. *SIGCHI Bull.* 29, 1, 32–35 (1997).

[3] A. Celentano and E. Dubois. A layered structure for a design space dedicated to rich interactive multimedia content. *Multimedia Tools Appl.* 76, 4, 5191–5220 (2017).

[4] A. Celentano, E. Dubois. Interaction-in-the-large vs interaction-in-the-small in multi-device systems. In *Proceedings of the 12th Biannual Conference on Italian SIGCHI Chapter, CHItaly '17*, 19:1–19:10 (2017).

[5] A. Celentano, D. Fogli, P. Mussio, F. Pittarello. Agents for Distributed Context-Aware Interaction. In *Proceedings of AIMS '02, Workshop on Artificial Intelligence in Mobile Systems*, 29–36 (2002).

[6] S. Chatty. Supporting multidisciplinary software composition for interactive applications. In *Proceedings of the 7th International Conference on Software Composition, SC'08*, 173–189 (2008).

[7] J. Coutaz. Architectural design for user interfaces. In *Proceedings of the 3rd European Software Engineering Conference, ESEC '91*, 7–22 (1991).

[8] F. Daniel, J. Yu, B. Benatallah, F. Casati, M. Matera, R. Saint-Paul. Understanding UI integration: A survey of problems, technologies, and opportunities. *IEEE Internet Computing* 11, 3, 59–66 (2007).

[9] M. De Paula, S. Barbosa, C. De Lucena. Conveying human-computer interaction concerns to software engineers through an interaction model. In *Proceedings of the 2005 Latin American Conference on Human-computer Interaction, CLIHC '05*, 109–119 (2005).

[10] F. DeRemer, H. Kron. Programming-in-the large versus programming-in-the-small. *IEEE Trans. on Software Engineering*, SE-2, 2, 80–86 (1976).

[11] T. Di Mascio, L. Tarantino, G.D. Gasperis. If usability evaluation and software performance evaluation shook their hands: A perspective. In *Product-Focused Software Process Improvement — 16th Int. Conf. PROFES 2015*, 479–489 (2015).

[12] A. Dix, , D. Ramduny, J. Wilkinson. Interaction in the large. *Interacting with Computers* 11, 1, 9–32 (1998).

[13] N. Elmqvist. Distributed user interfaces: State of the art. In J.A. Gallud, R. Tesoriero, V.M.R. Penichet (eds), *Distributed User Interfaces* Human-Computer Interaction Series. Springer-Verlag, London, UK (2011).

[14] H. Johnson, E. O'Neill, P. Johnson. Interaction in the large: Developing a framework for integrating models in HCI. In *CHI '99 Extended Abstracts on Human Factors in Computing Systems, CHI EA '99*, 165–165 (1999).

[15] W. Kuhn. Handling data spatially: spatializing user interfaces. In *Advances in GIS Research II, Proceedings of Seventh International Symposium on Spatial Data Handling* (1996).

[16] K. Kuutti.. Activity theory as a potential framework for human-computer interaction research. In B.A. Nardi (ed.) *Context and consciousness,*. Massachusetts Institute of Technology, Cambridge, MA, USA 17–44 (1995).

[17] K. Luyten, K. Coninx. Distributed user interface elements to support smart interaction spaces. In. *IEEE International Symposium on Multimedia*, 277–286 (2005).

[18] D.A. Norman. *The Design of Everyday Things.* Basic Books, Inc., New York, NY, USA (2002)

[19] D.A. Norman. Human-centered design considered harmful. *Interactions* 12, 4, 14–19 (2005).

[20] F. Paternò, C. Santoro. A logical framework for multi-device user interfaces. In *Proceedings of the 4th ACM SIGCHI Symposium on Engineering Interactive Computing Systems, EICS '12*, 45–50 (2012).

[21] F. Pittarello, A. Celentano. Interaction locus: a Multimodal Approach for the Structuring of Virtual Spaces , *HCITALY 2001*, 411–416 (2001).

[22] A. Seffah, M.C. Desmarais M.C., E. Metzker. HCI, usability and software engineering integration: Present and future. In D.M. Seffah A. Gulliksen J. (ed.) *Human-Centered Software Engineering — Integrating Usability in the Software Development Lifecycle*, Human-Computer Interaction Series, Vol. 8. Springer, Dordrecht (2005).

[23] N.A. Stanton. Hierarchical task analysis: Developments, applications, and extensions. *Applied Ergonomics* 37, 1. 55–79 (2006).

[24] A.G. Sutcliffe. Convergence or competition between software engineering and human computer interaction. In D.M. Seffah A. Gulliksen J. (ed.) *Human-Centered Software Engineering — Integrating Usability in the Software Development Lifecycle*, Human-Computer Interaction Series, Vol. 8. Springer, Dordrecht (2005).

[25] J. Vanderdonckt. Distributed user interfaces: How to distribute user interface elements across users, platforms, and environments. In *Proceedings of XI Congreso Internacional de Interacción Persona-Ordenador, Interacción'2010*, 3–14 (2010).