

# Chapter 10

## Interaction Design at Itsme

Giorgio De Michelis

### 10.1 Introduction

After a 1 year long preparation, at April 1st 2008, we had the kickoff meeting of *itsme*, a project with the ambition of designing and building an innovative front end of Linux for workstations (De Michelis et al. 2009). The idea behind our project was to go beyond the desktop metaphor shaping all existing operating systems for workstations (Windows, MAC OS, Linux versions like Ubuntu, etc.) to create a new system able to support the context awareness of its users.

Our project is still ongoing (after a radical stop, due to the end of the consumption of the first round collected investment, we have slowly restarted to work in a two persons' team, and we have completed in these days a first running prototype exhibiting the main features of our system, and we are now planning to show it to a new group of investors, to move toward the market), but it makes sense, I think, to reflect and to discuss with a sensible audience what we have learned about how it is possible to design a radical innovation in a crucial sector of ICT, where there is a "de facto" standard and billions of people use it all over the world.

The issue I want to bring to the attention of the readers in this paper is how we designed *itsme*, with a particular attention on how we inflected interaction design in the process and on how we tried to interact with and get feedbacks from the users even before they could test and evaluate our system. We have used in our project unconventional versions of interaction and participatory design, but we did it paying a constant attention on how we were working, and therefore, we learned a lot on design of systems for the general public like an operating system for workstation (or its front end). Our attention on how we were designing was motivated, not only by

---

G. De Michelis (✉)  
DISCO, University of Milano – Bicocca, Milan, Italy  
e-mail: [gdemich@disco.unimib.it](mailto:gdemich@disco.unimib.it)

my academic interest on design but also and mainly by my awareness that designing a successful innovation, capable to substitute a widely diffused standard, needs a maximum of cure.

The paper is organized as follows: first, we shortly recall how we arrived to decide to try to design a radically innovative front end for the operating system of workstations, and then we discuss the main steps in our project. We will survey in particular: the composition of the itsme concept, the move from the concept to the software architecture, the interaction with the users, and the changes of the concept we did while design was still ongoing.

The last section will be dedicated to a discussion of our view on the couple “innovation-continuity,” showing that, paradoxically, disruptiveness softens the opposition between them.

## 10.2 Beyond the Desktop Metaphor

In 1984, Apple, inspired by Alto and Star workstations of Xerox, presented Macintosh (in short, Mac) its innovative personal computer. The Mac had a great impact on the market and greater one on the companies producing personal computers and/or their operating systems, defining a “de facto” standard characterizing the features of all the workstations that would be offered on the IT market in the future. Graphic interfaces, mouse, windows, and icons can be considered the most innovative features characterizing the Mac, but what is mostly relevant is that all those features shaped the interaction with the machine on the basis of a simple and intuitive metaphor. As proposed by the Xerox PARC researchers, the workstation interface is based, in fact, on the “desktop metaphor” (Kay 1977), i.e., a replica of the surface on which users work: a plane supporting and containing several distinctive tools, documents, and objects of different types, a hierarchy of folders, a trash basket, etc. The success of the desktop metaphor stems from its ability to reproduce the arrangement of tangible things: if the personal computer is a multifunctional machine allowing to perform several distinct tasks, then users can easily do their tasks on it in the same way as they do on their desktops, without needing specific training and, even, without looking at their manuals.

As we have recalled above, in few years, all workstations adopted the same approach for their interfaces (Windows1 appeared at the end of 1985 with a limited success, but Windows2 that appeared at the end of 1987 gained in short time an immense popularity and a great share of the market), making the “desktop metaphor” the “de facto” standard for personal computers. However, the technological evolution of workstations after 1984 (always larger internal disks, e-mail becoming a universal medium, the World Wide Web with its growing information, services, and social interaction) has dramatically increased the number of tasks supported by PCs, adding, in particular, content and information management tasks to those that were already supported, and the digital contents and information

accessible to their users grew even more rapidly (Eppler and Mengis 2006). Operating systems for workstations have tried to accompany these changes increasing the number of their functions and services, but this has made them always more complicated and difficult to use, while users have slightly changed their way of using personal computers in directions that are always more distant from their original nature of multitasking tools.

Surveying a literature dating back up to the early 1990s, Victor Kaptelinin and Mary Czerwinski, the editors of *Beyond the Desktop Metaphor* (2007), sum up that “the development of desktop systems over the last two decades has revealed limitations of the desktop metaphor. In particular, the metaphor does not provide adequate support for the access to information objects along with the display of the content of those objects, multitasking, dealing with multiple information hierarchies, communication and collaboration, and coordinated use of multiple technologies.” (ibidem, 6).

It is difficult to dispute the above claim, since, even as users, most of us (I imagine to share with my readers a life of “knowledge worker”<sup>1</sup>) have experienced the limits listed there. The authors of the above quoted book continue their intellectual exercise giving form to what should substitute the current organization of personal computers claiming that metaphors themselves are inadequate for driving the design of new ways of interacting with digital devices and content (Freeman and Gelernter 2007, 23; “We prefer to approach software design not by metaphors but by Nelson’s concept of virtuality”). The limitations of the desktop metaphor appear to them as due to the increasing number of tasks supported by present-day personal computers, in particular on the content management side and to their inability to deal with the social dimension of human tasks.

When, on my side, in 2007, I begun to think how the limits of the desktop metaphor could be overcome, I agreed with the above critiques, but I was convinced that I needed also to go beyond the task-based view of personal computers to better capture what was happening in their use: users, more specifically some users, in fact, were changing what they were asking to ICT, from being supported in a large variety of tasks related with information management to having a companion in all the circumstances of their life. Instead of abandoning “metaphors” for “virtuality,” I tried, therefore, to look for a new metaphor capturing the changes undergoing in the way people use their PCs.

The distinction of some PC users among all that I did in the previous claim should not surprise the reader: in accordance with what has happened with other technological products (from pens to cars), becoming a universally usable machine has induced a differentiation process among its users. Even if a serious analysis of the emerging profiles of PC users has not yet been accomplished, we can recall that, today, users of personal computers encompass, on the one side, those using it, almost exclusively, to Skype with other people and for other simple tasks (e.g., the parents of sons and/or daughters studying away from home), on the other, people who are progressively becoming unable to do anything without the support of their

---

<sup>1</sup>In the next pages, we will come back to knowledge workers.

personal computer (e.g., knowledge workers). Without spending further words on the emergence of different user profiles, let us concentrate our attention on the people for which the PC is an unavoidable tool and on the way they use it.

As I have said above, I am thinking of active knowledge workers,<sup>2</sup> who are engaged in a variety of endeavors where they concur to create new knowledge and to publicize it in (multimedia) documents: for all of them, personal computers have become unavoidable, but always more inefficient, tools. The fact that today's personal computers contain a relevant part of the content and information they need and the tools for navigating and manipulating it, as well as almost all the knowledge they create (in the form of messages, posts, and multimedia documents) and the tools for creating it, modifies the way knowledge workers interact with their personal computers. It is not, generally, a matter of being supported in managing information, since searching, classifying, creating folders, etc., are time consuming and distracting tasks, with respect to the practice triggering them: what users want and need is having ready at hands, on the screen of their personal computer, all what is relevant for what they are currently doing. Combining the management of information with the execution of tasks, in fact, the way knowledge workers use their PC, is radically changing: most of them switch it on, as soon as they reach their workplace, because it has become an extension of their capability of acting and interacting, since, in most situations, only in accessing the information stored in the PC can they perform effectively. But while the task-oriented organization of their personal computers asks them to retrieve the content, using more or less sophisticated searching mechanisms, they need it immediately "present at hand," without distractions and losses of time. We, CSCW researchers and practitioners, know what this change in perspective means, because we have studied it from the very beginning of our common endeavor: recognizing the situatedness of our action and interaction in a context that is not only spatial and temporal, but, most importantly, social.<sup>3</sup> When we say that human action and interaction are situated in a context, in fact, we may refer to diverse dimensions of it: *spatial* context (where a person affects what she can do), *temporal* context (time is deeply affecting and/or conditioning what a person does), and, finally, *social* context (the people participating in a common experience share knowledge of their past,

---

<sup>2</sup>Knowledge workers (Blackler et al. 1993) have emerged as the most important category of workers within offices in the last 30 years. In the preface to *Knowledge workers in the Information Society*, Mosco and McKercher (2007) recall three main definitions of knowledge work. The first one, and most narrow, considers knowledge work any practice involving "the direct manipulation of symbols to create an original knowledge product or to add obvious value to an existing one." The second that extends sensibly the previous one considers knowledge work any practice involving the management and distribution of information. The third that is the broadest one considers knowledge work any practice involved in "the chain of producing and distributing knowledge products." People doing knowledge work under the broadest of these definitions correspond to our profile of PC users.

<sup>3</sup>Situatedness has been one major theme of research in CSCW from its very beginning. Without any aim of completeness, we can remember: Suchman 2007; Agostini et al. 1996; Schmidt 2002; De Michelis, Chap. 5, this book.

mutual commitments, and their potential for action and interaction). The social dimension of context is what matters in these pages, since it permeates any action and interaction of a person, imposing him/her to go beyond their tasks.

Without deepening the discussion on situatedness, I spend few lines for discussing the systems allowing people participating in common experiences to share knowledge. Early groupware systems – from the Coordinator (Winograd and Flores 1986) to Notes (Kawell et al. 1988) and beyond (e.g., Google Wave (Siegler 2009) and Google+ (Sanjay Kairam et al. 2012)) – cannot fully solve the situatedness problem, since they create and rely on spaces for collaboration among different users that are separate from their individual workspaces. Services like shared calendars and spaces, document management systems, networked repositories, co-browsing tools, and chats have all enriched what workstations offer to users, but none of them is able to take into account how the social experiences people are sharing influence their individual experiences, enriching and/or affecting them. This is visible only if we take into account that social contexts appear different from the viewpoints of their diverse individual participants, i.e., if we deal with them at the diverse individual workplaces of their participants.

It is the consideration of situatedness, or, in other words, context awareness, from an individual viewpoint, the issue that guided me toward inventing a new front end of a personal computer or, more precisely, of its operating system (called *itsme*). There is a growing evidence that the complexity users meet when trying to get rid of the large amount of objects stored in folders is not purely quantitative, since users at any moment need to access only those documents that constitute the pragmatic and semantic context of their current focus of attention. They have no ways to select what interests them in a given moment among all their objects. The context awareness provided by current workstations, let me repeat it here, is limited and partial: productivity tools (even if collected in suites) still have supremacy over the usage context of their objects, and, moreover, there is a clear-cut separation between them and communication and/or Web browsing packages, not allowing to organize all objects on the basis of their usage context.

### 10.3 The Itsme Project

The starting point of the *itsme* project has been the discovery that going beyond a task-based organization of ICT systems is not possible at the level of applications that are distinct and function-oriented pieces of software. Context awareness is not something users need while they are working with specific applications: on the contrary, they need it, irrespectively of the application they use, to articulate their work in relation with other people, to organize themselves, to access relevant pieces of knowledge, and to know what to do next. The services supporting context awareness should glue all the applications a person is using, so that all and only the knowledge related to a context are accessible altogether.

We need, therefore, to modify existing operating systems embedding in them context awareness services. This risks resulting in a greater complexity of workstations at the interface level, if we don't change them in a radical way, applying the simple rules governing contexts. As said in the previous section, contexts continuously form.

In order to provide users with an integrated experience of the services we have designed for context awareness in these years, we need to design a new (operating system for) workstation where, abandoning the desktop metaphor, what a user does is embedded in the context giving it sense.

Reflecting on the research done within the CSCW community, we conceived a new metaphor for the "look and feel" of a workstation, namely, that of *stories and venues* (please note the plural!). Everything users do is in the context of one of the diverse stories they live with other people. Any of these *stories* has created or imported during its evolution a large variety of things (digital and/or physical objects of different types, people's addresses, relevant URLs, exchanged messages, etc.), and, to perform adequately within that story, users need to have them, ready at hand, in a unique (physical and/or virtual, augmented) place, in a unique *venue*.

This requires, on the one hand, that the system goes beyond the distribution in different places of the things people need in their activities determined by the applications currently installed in a workstation (the documents created by the user are stored in the file system, the e-mail messages are inside the mail client, their attachments are in a particular folder of the file system, the bookmarked Web documents are elements of a list in the browser, etc.) putting them together.

On the other hand, stories are partially reflected in the threads generated by user interactions.<sup>4</sup> A story can, therefore, be seen as a bunch of threads of interrelated heterogeneous events (from the viewpoint of the applications supporting them: e-mail and chat conversations, RSS feeds, posts in a social computing platform, etc.). The creation and maintenance of the related venue can be straightforward and do not need user intervention, but corrections and refinements, if we let the system keep together its threads and add to a venue any new thread originating in it. By the way, it has to be underlined that this does not require that all the interacting users have workstations with the itsme operating system (i.e., based on the venues' metaphor), since itsme can locate objects in venues using only the information characterizing threads.

---

<sup>4</sup>After the coordinator, presented by Fernando Flores and Terry Winograd in 1986 (Winograd and Flores 1986), there has been a rich debate in the CSCW community following two different directions: on the one side, Lucy Suchman (1993) discussed it for its unnatural forcing human conversations within formalized patterns, allowing hierarchical control on it; on the other side, several authors paid a growing attention to conversations as threads of communication events underlining their switching among different media (Reder and Schwab 1988, 1990) and showing their relevance, beyond their reduction to illocutionary acts (Bullen and Bennett 1990; Winograd 1994; De Michelis and Grasso 1994). After a period where attention on them declined, threads have gained again attention with the Google Wave and Google+.

In our previous research, we revisited the language/action perspective (De Michelis and Grasso 1994) claiming the importance of threads, and these same concepts are now the bases of the new metaphor we are proposing:

- A thread contributes to constitute a context and its venue.
- Venues aggregate threads and the related objects.
- New venues are created as new threads begin, either from scratch (i.e., by reacting to an event) or spinning off existing ones (e.g., when the issue characterizing a thread changes or multiplies).
- Users can modify venues sorting the objects they contain, merging or splitting different venues, deleting them, and so on.
- Objects only exist within a venue or, when a venue has not yet been created, in a special space, called *limbo*, where all threads and objects not yet assigned to a venue are stored.
- A thread, and/or an object, can be stored, through replicated references, in several, diverse venues, if it contributes to constitute diverse contexts.

While the desktop metaphor does not preserve the individuality of different contexts, meaning that users can handle objects related to different contexts at the same time, in the venues' metaphor at any time, only one venue is open in front of the user. All venues are contained in the "home," where the user is brought when he/she switches on: the home is where he/she can browse all his/her venues and open any of them.

In itsme, I can, with few clicks, create a venue for each of my stories and manage them (with no constraints: I can create few – up to zero – or many stories, small or big stories; I can create them early or later, when they are well consolidated leaving the relative threads in the limbo; I can split a story in two, or merge two stories; I can move an object or a thread between two venues; etc.). Venues are self-updating, even if they are not "intelligent," because itsme uses metadata to keep threads together.

A quick example may help to put together the information presented above: like most of my peers, my life is complicated, due to my many commitments and engagements. During a typical working day, I switch from my lectures to the paper I am writing with two colleagues, from the master thesis in which I am advisor to a meeting of the foundation whose board I am a member of: all these are, in itsme terms, stories. Whenever I do a switch between two stories, I need to quickly situate myself in the new one, in order to be able to act and interact effectively in it: having a venue for each of my stories lets me have all what I need ready at hand. Consider the following case: while I am reading a paper, I receive a telephone call from my colleague Giuseppe asking me if I have read the document he sent me. He needs my comments, before going to the meeting where he will present his document. I received his document some weeks ago, and I have a vague recollection of it: it seems to me that I have read it, but I can't remember if I have some written notes on it or not. With itsme, I go to venue related with the story I share with Giuseppe, and there, I find his document and, if they exist, my written notes on it, so that I can almost immediately react to Giuseppe request; without itsme, I should search for Giuseppe's document and for my notes, if they exist, and it is not certain that this

search will be straightforward. Itsme is not commenting Giuseppe's document for me, it is not doing my work, it only helps me to concentrate on my work without spending time in searching and recollecting!

Technically speaking, the starting point for itsme is the Linux operating system. In fact, itsme is going to be deployed as a Linux distribution with a peculiar front end that will exploit a set of characteristic services. Its novel graphical user interface relying on and illustrating the relationships between objects that characterize venues is supported by an intermediate stratus of software, allowing to associate to all the objects stored in the file system metadata and links and to manage them. The use of Linux will allow for the exploitation of all the available components, and itsme will represent a radical innovation and carry a significant contribution (in terms both of technology and of the presence of practitioners) to the existing community. However, the affiliation of itsme to the open source scene goes beyond the adoption of Linux development as a platform: one of the slogans with which the itsme concept has been demonstrated is *let's team up*, meaning that the involvement of people in the project is perceived to be essential. This idea basically translates to three aspects:

1. The constitution of a community following and providing directions for the project, implementing a form of participatory design; a great effort is being put in creating a "real-life" community of people meeting each other (through seminars and participation in public events), and also some of the so-called Web 2.0 services, such as social network sites, are being employed to make the community constantly grow and live.
2. The search for contributions from outside the company: new ideas, concepts, visual design components, and code. Community members are also being involved in the evaluation of ideas and the validation of design products.
3. Community members are helping us to generate hype on what we are doing and on the (long disregarded) issues that we are trying to solve. People are helping us to create the market before the real workstation is ready to be sold. In this way, itsme candidates itself to be the fulcrum for the innovation in consumer applications such as productivity tools and Web-based and e-mail-based applications.

The development of itsme has been carried on, up to now, in two phases. In the first one (2008–2011), a design and development team of almost 15 people worked full time on the itsme prototype. In 2011, the resources granted to itsme by its business angels came to an end, but the new resources we expected from private (venture capital) and public funding did not show up against our expectations. The project group was dissolved, and it started a new phase (2011–now) where the project continued with the aim to complete the first prototype release of itsme, even if I could only rely on one part-time developer (who was part of the team working on itsme during the first phase) and some students. The prototype, with an improved interface, has been completed in February 2014 together with a new release of the concept manual, and we make reference to the latter in our description of the system. We plan now to show the prototype to potential investors, to get new resources for



the final steps toward the delivery of itsme to the market. During this third phase, we will reanimate the community supporting our project providing to its members a way to access and test the prototype. In this way, the development of the full version of itsme will reflect their critiques and suggestions.

## 10.4 The Team

The team engaged in the first phase of the itsme project merits some words. Besides me, there were 16 young people with different professional backgrounds: almost one half of them were software specialists, and their role was mainly to design and develop the software; 5 people were industrial and/or graphical designers, and their role was both interaction and interface design; the rest had different backgrounds (philosophy, business design, communication) and were managing the communication of the project (website, presence at fairs, relationships with the press, etc.). I paid a great care to create out of them a well-amalgamated team, so that each of them felt to fully participate in the creation of an innovative piece of software with a potential great impact on the ICT sector. I tried to get this through different means: frequent meetings of the whole team to discuss design choices at any level, mixing people with different backgrounds in the small groups dealing with specific tasks or responsibilities, and seminars dedicated to reflect on the difficulties and risks of innovation design. Seminars have been dedicated to issues like how to find a balance between innovation and continuity, so that people who already use existing personal computers can move to itsme without effort, being prepared to redo all what we have done up to a certain moment, or a large part of it, when we discover that better choices are possible. They played a big role in creating an identity in the team.

I tried also to keep the team as open as possible, letting its members to participate in meeting with external people from the business, the technical and the design field, and constantly inviting them to discuss what they are doing with any person showing interest in the project.

Openness of the team is the first step toward being able to listen to users whenever it is possible. And users have been a constant concern during our project.

## 10.5 The Concept

When we did the kickoff meeting, at April 1st 2008, I had a clear idea of the “stories and venues” metaphor and of what itsme should be, but most of the team didn’t know so much about it: they had only the vague idea of the new system they got from their conversations with me before being enrolled in the project. We decided to begin our work in accordance with two guidelines: (1) we wanted to adopt what could be considered, at best, an interaction design approach (De Michelis 2003),

and (2) apart from the generic features of the system we had to design and develop (a Linux-based front end, the “stories and venues” metaphor, a system together with the services users could need for adopting it easily), nothing was fixed.

In few months, we came out with the first thing of our project: the concept manual. It was a rich description of the “look and feel” of the system and of the main features<sup>5</sup> characterizing it: the venues, the transit, the limbo, etc.

The effort we did for writing the first version of the concept manual involved mainly the designers of the team, but it was constantly widespread in the whole team, so that the concept became the document around which the object of design of our team was being created and shared and our communication policy was taking form. After the concept manual, we moved quickly away from the open brainstorming phase, to pass to a more focused design phase, where all what we did was put in relation with the concept. In particular, the concept was of paramount importance for both the first user participation phase and the design of the specs of the software we had to develop.

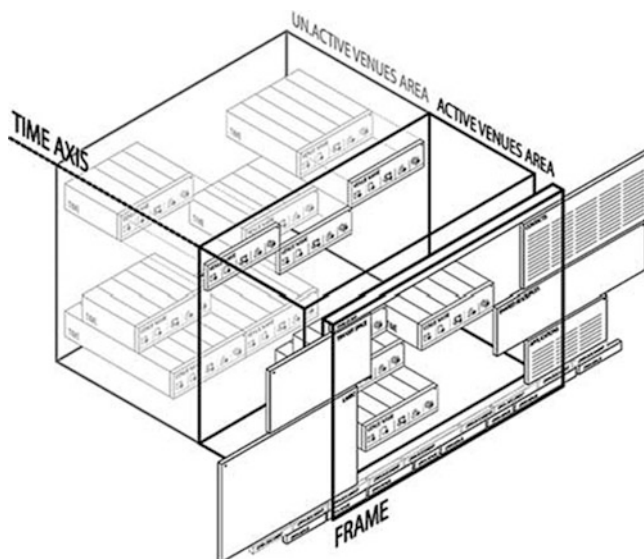
Our design process aimed at transposing the conceptual framework of itsme into proper design requirements and specifications for the operating system. It has to be underlined that the concept manual of itsme is not a full specification of the system, but the definition of what we are designing. The concept is continuously evolving in parallel with the design and development of the system without a strong alignment between them: in some sense, the differences between the concept and the system testify a design tension always present in the team and grant its openness to changes. The design activity has alternated phases dedicated to the investigation and development of specific design topics with phases dedicated to the refinement of itsme conceptual framework and assumptions and to the exploration of technical requirements and implications. More in detail, the process has followed a macro plan, starting with a problem setting phase, followed by the definition of an interaction design model and the exploration of interaction paradigms and a preliminary graphical user interface through schematic representations, to achieve a detailed design of the look and feel of the itsme visual interface. Given the complexity of the general task, the process has continuously evolved and adapted: a number of micro-activities and workshops involving both the design team, potential users, and the community of interest and stakeholders were organized in relation to the emergent needs.

From an interaction design perspective, itsme can be defined as a dynamic, adaptive, and supportive system whose dialogical qualities are emerging in relation to the interaction with, and by, its users.

While supporting the users in the creation and sedimentation of their personal stories, itsme manifests its own identity, acting more as a partner than as a tool, providing to the user both the context and the rationale for the management of documents, applications, and resources.

---

<sup>5</sup>We will give more details on the features of itsme in the next section.



**Fig. 10.1** The itsme interface architecture

From a structural point of view (Fig. 10.1), the interface of itsme is a continuous space where the x and y axes correspond to the visualization area of the screen, while the z axis corresponds to a timeline. Venues are automatically placed along the time axis, following their chronological order, but can be freely organized by the user on the screen surface that, as a sort of digital trompe-l’oeil, displays recent and active venues.

The *home* (Fig. 10.2) of itsme is, therefore, populated by the venues. In the home, venues may have two representations (Fig. 10.3): one, highly synthetic, is a stylized symbol with the short name and a number, indicating how many new objects are present in the venue, while the other, more detailed, indicates also, for each type of objects, the number of items and of new items contained in the venue. In the home, like in any other visualization of itsme, in the upper right angle, there is a button with a “plus” symbol, for the creation of a new venue.

From the interaction and interface point of view, *venues* represent the main contexts for personal information and content management: as said above, a venue (Fig. 10.4) is the place where the user finds all what is relevant within one of his/her stories and can manage his/her personal information and communication flow within it.

Within a venue, there are different channels, each represented by a specific tab. Each channel is both a way to easily access a specific category of objects and to quickly create new objects in that category: *messages* contains e-mails and chats, *documents* contains editable files, and *media* contains audio-visual read-only files. The right part of the venue contains the resources in use in the related story: *people*, *Web resources*, and *applications*. One of the main features within a venue, based on



Fig. 10.2 The itsme home

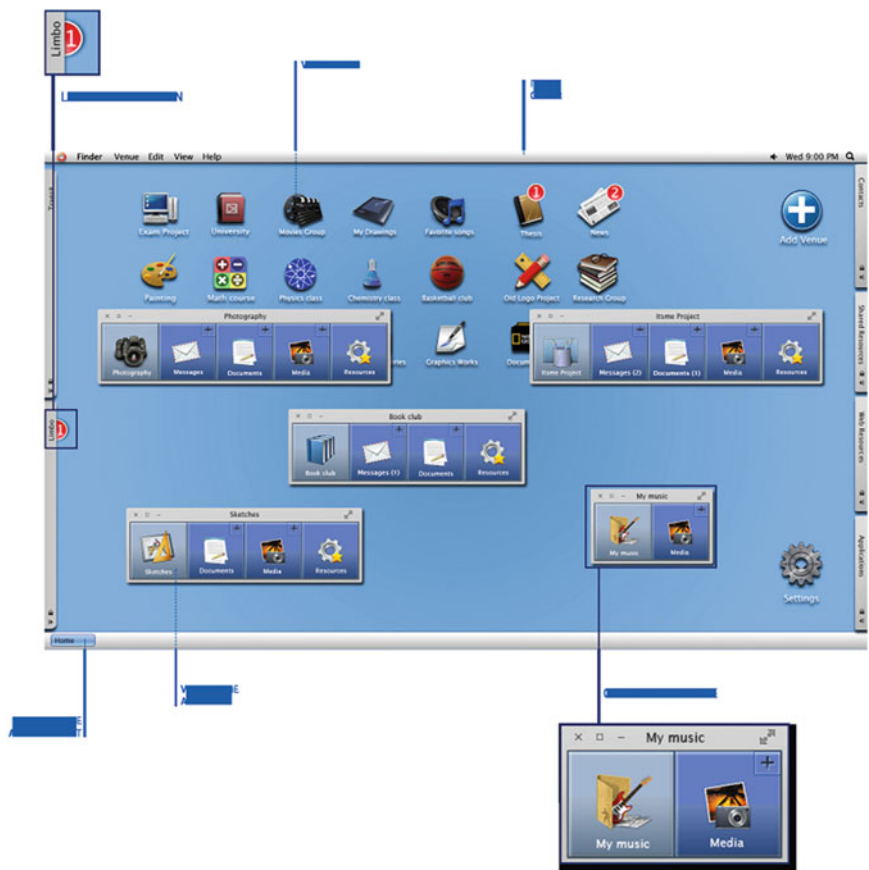


Fig. 10.3 More about the itsme home



Fig. 10.4 A venue

the use of metadata, is the visualization of the correlations between different files activating a “highlight” mode that, starting from a specific object (e.g., an e-mail message), shows the whole thread containing it (e.g., the whole conversation, all its attachments, and the contacts to all the people participating in it as senders or recipients).

As underlined in Fig. 10.1, the frame surrounding what is displayed on the screen plays a relevant role from the interaction viewpoint.

Its vertical bars contain some sliding menus (Fig. 10.4). Those at the right bar allow a direct access to all the resources of the workstations: contacts, shared resource and web resources, and, finally, applications. More in detail:

- The *contacts* menu allows the access to the whole address book of the workstation.
- The *shared resources* menu allows access to both the *mail* (i.e., a standard e-mail client for Linux) and the *agenda* of the workstation (Fig. 10.5).
- The *Web resources* menu allows access to all the bookmarked informative resources.
- The *applications* menu allows access to all, local and remote, applications.

On the other hand, at the left bar, there are two sliding menus: transit and limbo:

- The *transit* menu allows the access to a buffer containing all the objects that are waiting for being located in a venue. There is a time limit for the permanence of these documents in the transit, after which they are moved to the limbo.
- The *limbo* menu allows the access to a space (resembling a venue) containing all those objects that are not located in a venue, i.e., that do not belong to a story, yet.

# ITSME PROJECT VENUE AGENDA CONFIGURATION

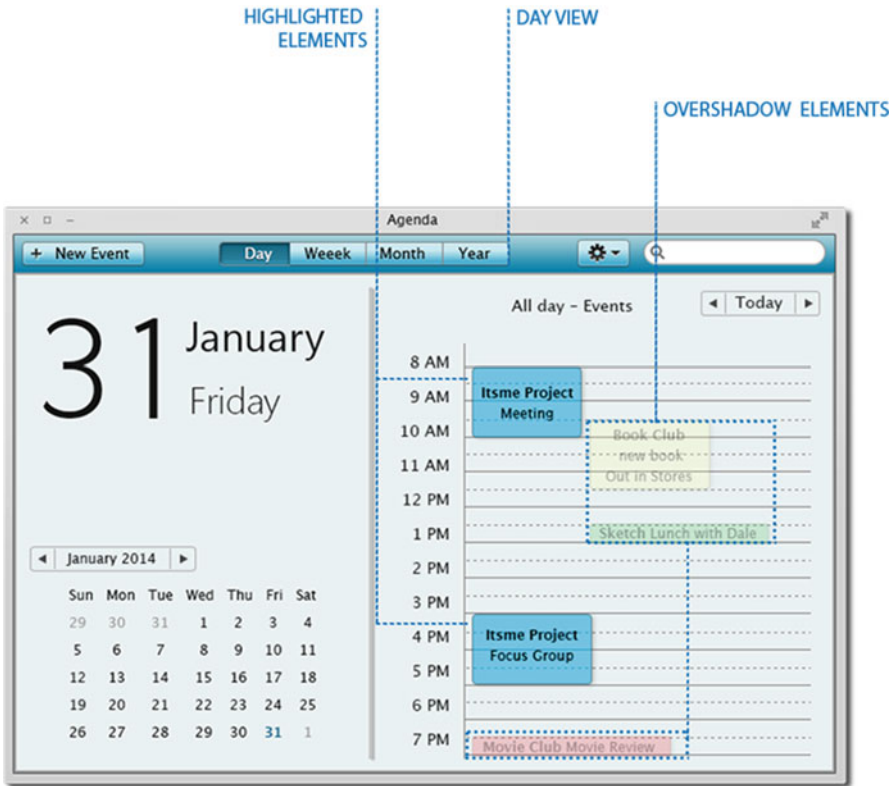


Fig. 10.5 The agenda of itsme

If and when a user has not yet created any venue, the limbo contains altogether all the objects of the workstation, avoiding their distribution in diverse places as with current operating systems (Fig. 10.6).

The two horizontal bars have been inspired by the interfaces characterizing most browsers (e.g., Firefox and Chrome) today: the upper bar contains the visualized venue, and the objects open in it; the lower one contains the home of itsme and the open venues.

This means that, if three documents are contemporarily open in a venue (Fig. 10.7), then they are also listed in the upper bar. If the user clicks on another venue in the lower bar, then the display substitutes the previous venue and its open documents with the newly selected one with its open documents, and the same does on the upper bar.

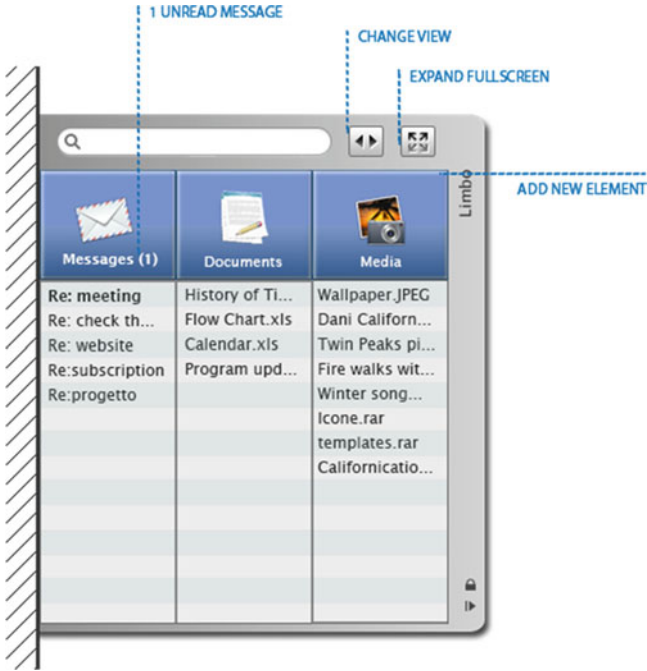


Fig. 10.6 The *limbo* menu

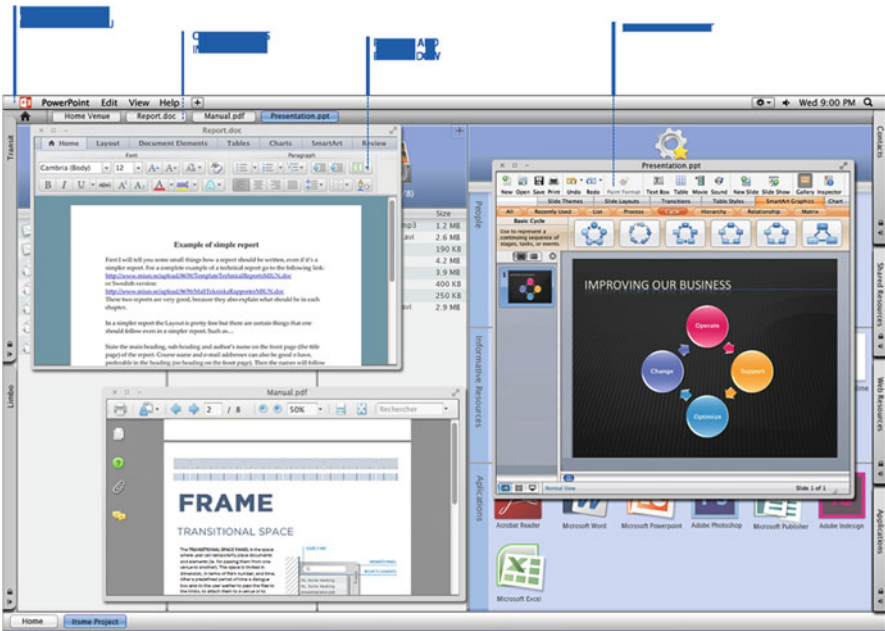


Fig. 10.7 A venue with the documents open in it



## 10.6 An Open Source System

All existing operating systems for workstations are based, as we said above, on the desktop metaphor. But this is not the only thing that they share: in fact, they share also a file system whose elements are just files of bits. But, for creating and managing venues, we need a smart file system whose objects can be tagged and linked, so that we can situate each of them in the right venue and we can browse threads. Both the fact that we are developing a new operating system whose objects are qualified by metadata and the fact that we want to develop it in collaboration with the users, listening to their needs and desires and conversing with them on our design choices, has brought us to choose the Linux operating system as the basis of the system implementing the “stories and venues” metaphor. This choice has several advantages: with respect to building a new operating system from scratch, it allows to concentrate the effort on the front end of the system and on the layer connecting it with the Linux core (whose role is allowing the creation and management of metadata decorating objects) since its openness allows an easy development of new features and functions on top of it; with respect to building it as an extension of a proprietary operating system, it allows to keep the openness and interoperability of our software at its maximum, granting to our users the capability to interact with other people who are not using itsme and to develop it within the large open source community, discussing within it all the issues arising from our design and development process and trying to push innovation in Linux (e.g., augmenting its file system with tags and links) and in its applications (e.g., office suites, interaction protocols, etc.).

As it appears clearly in Fig. 10.8, the architecture of itsme consists of two layers on top of Linux interconnected among them through a standard *proxy*: the *user interface* (in orange) embedding the desktop metaphor and the smart extension of Linux (in green) where the *logic* (the metadata and item managers, in particular), making the Linux file system smarter (its files have tags and links), is the basis for the *interface* (the event notifier and other modules necessary to react to events) being able to put objects in the venues.

## 10.7 Users Before Usage

Once the concept manual was in our hands, even if we didn’t have any piece of software visible, we decided to interact with potential users as soon as possible to verify if what we had conceived was capable to meet their needs and desires. Moving from the metaphor of “stories and venues” to the look and feel of itsme, we did not have, in fact, any certainty neither that our discourse fitted with user needs and desires nor that itsme could be considered a decent candidate for satisfying them. We started, in those first months of work, to live the experience of being in a crystal bowl, detached from the real world, where internal coherency of what we were doing could take over dramatically on external valuation by the future (potential) users of the system.



# itsme OS architecture

as in the current prototype – November, 2009

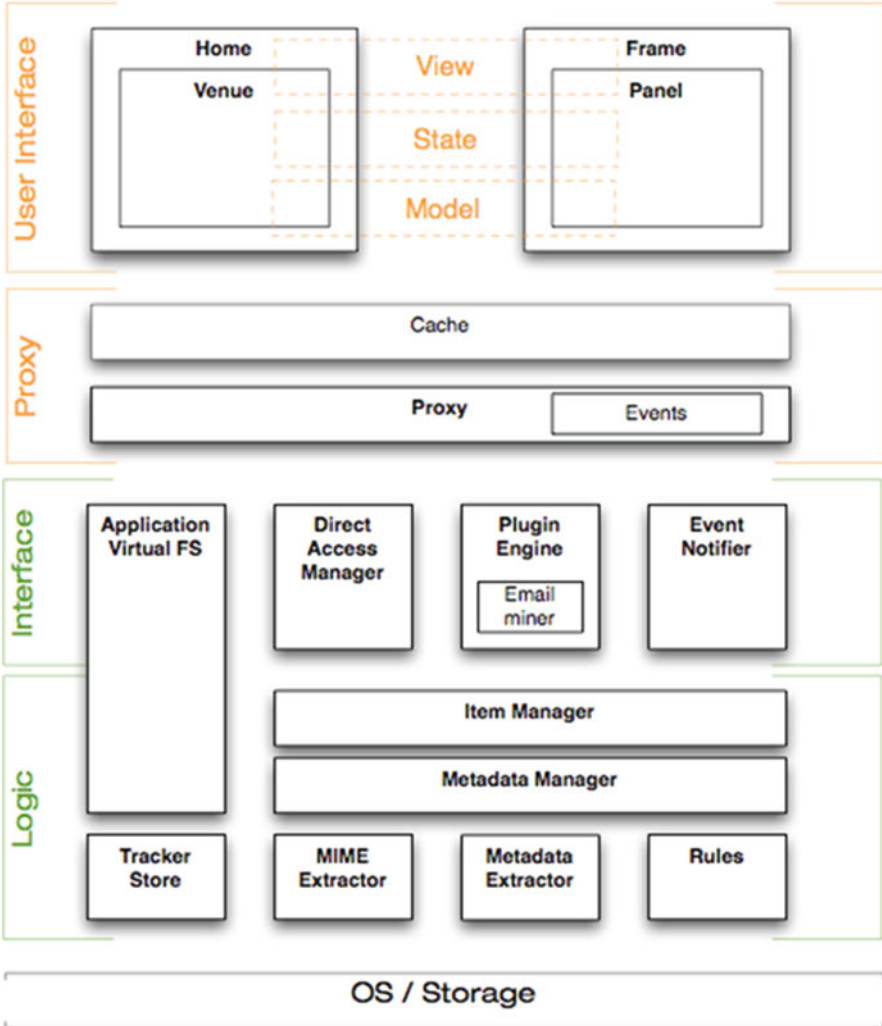


Fig. 10.8 The architecture of itsme

I want to stress here that listening to users was an urgent need for orienting our design work, but also that there were no experiences that we could follow in order to do it effectively: participatory design, in fact, approaches have never been adopted for systems for the general public with the complexity of an operating system! Even

if we adopted in some way the Apple approach (Isaacson 2011), sublimating the personal experience of the designers themselves and of their relatives and friends, enriched by the knowledge about cooperative work emerging from 20 years of research in the CSCW field, we were absolutely aware that the radical innovation, we wanted to bring forth, needed to rely on well-conducted experiments involving real users.

We decided therefore to do a large (at least for our dimensions) experiment with real people, representing a reasonable sample of our target users. With the help of a team from the University of Siena, lead by Patrizia Marti, we prepared four short videos presenting four users solving intricate problems with itsme: a journalist gathers in a new venue all that she needs (papers, documents, Web resources, contacts, etc.) to write an article; a researcher browses past venues for finding interesting people for the workshop she is organizing and invites them; the owner of a travel agency finds a previous plan of a trip to Norway to use it as the basis of a new one and discusses it with colleagues and customers; a manager rearranges his agenda, when one of his appointments is anticipated. After having seen one or more videos, participants answered to a questionnaire that combined the evaluation of the potential utility of itsme with the explanation of it (Fig. 10.9). This evaluation was made of two parts: on the one hand, participants selected a comment on the video in a predefined set and on the other associated to it a word, to be immersed in a tag cloud.

The videos were shown to some hundreds of people, in two large Autumn events in Italy: first, in Milano at SMAU (the largest ICT fair in Italy), where participants were mostly young people passionate with technologies, and second, in Firenze

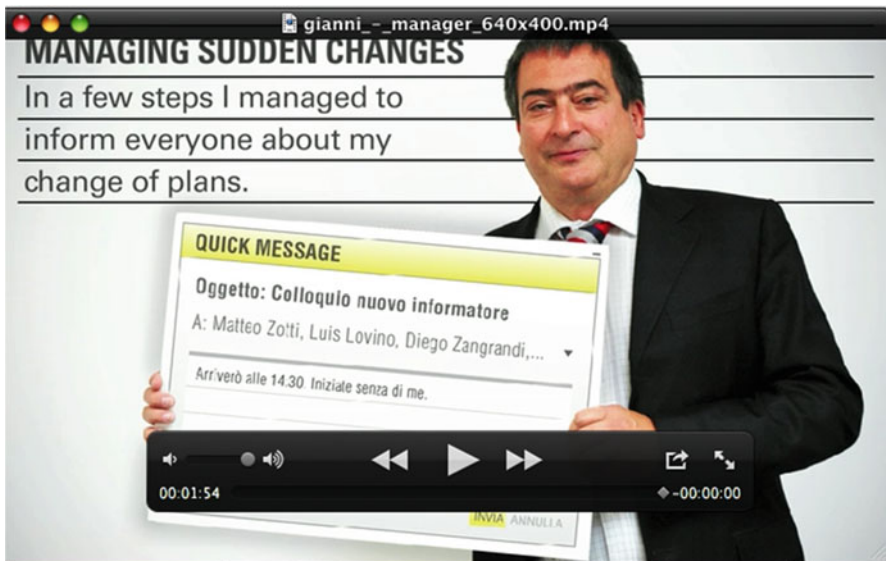


Fig. 10.9 A screenshot of one of the videos

at Creativity Festival (a big event on arts, design, hi-tech, and social animation), where participants were mostly young creative people. Our sample was, therefore, made of people well oriented with respect to innovation but with heterogeneous backgrounds, being good representatives of what could be considered the target of early adopters of itsme. The answers of the participants were quite positive with respect to the “stories and venues” metaphor and gave us some interesting indications about the reasons why they considered a system based on it useful that confirmed the rationale of our project (among the most appreciated features of itsme, having all what is relevant in a story in a venue, the time bar, the previewing system, etc.). The tag cloud emerging from the survey underlined our intuition that stories were a quite natural concept for looking to personal social experiences and that navigating among stories was considered as a strong improvement with respect to navigating among messages or documents. Both the participation in two large public events with our stand and our t-shirts and the success of our testing experience gave to the whole team a greater awareness of what we had to design and develop.

While the team was developing the prototype of itsme, always in order to get new feedbacks from the users about its qualities, we decided to develop, in parallel, a Web-based emulator of it. It was a Web platform providing to its users a personalized space where all the actions and interactions constituting a story were put together. Its interface was replicating the design we had defined for itsme, and we thought that it allowed users to experience partially how they can use a system supporting their stories.

The first tests we did with the emulator were quite controversial, with severe negative outcomes and few interesting, positive comments. The messages we got were so strong that we did not go on with a larger experiment, and we immediately reviewed our project on their basis.

The first mistake we did, presenting the emulator of itsme to some people (mainly friends, already interested in our project), was to put them in front of a void system: they could see the structure of venue, but there was not anything inside it. This caused a dramatic paralysis in the users: they did not know what they had to do but also what they could do. So, instead of getting responses about the usability of venues, we got a strong negative feedback about its understandability. Reacting to this unforeseen impasse, we filled the emulator with some generic content, and we made some new experiments with users. This time, we had two different reactions: users with ICT competence could understand what the emulator offered, at a meta level (their behavior was in fact that of experimenting the potential for action and interaction of venues, without any reference to what they need and desire in their everyday life), but the same did not happen with users with no or little ICT competence. Again, we had a very negative feedback: people did not understand what the emulator of itsme was for. The reaction of non-ICT professional users to the emulator needed an explanation: we found it in the fact that filling a venue with generic content could not make visible to users what a venue was for – supporting a story they are living – but looked like a generic aggregation of content and services, whose sense was not clear. We decided, therefore, that the only way for

presenting the emulator (and the same would become true for itsme, when ready for a beta test) was offering to users the possibility to access a story, corresponding to his/her habits, profession, and interests. We concentrated in professional stories that seemed to us having some “typical” features that could be embedded in a self-explaining story model: the main stories for general practitioners are those they live with their patients; the main story of a designer is his/her projects; the main stories of business angels are the start-ups on which they have invested; etc. We made, therefore, together with Carla Simone and Federico Cabitza of the University of Milano-Bicocca a study characterizing the care general practitioners give to their patients in terms of cure stories (Cabitza et al. 2014). It was surprising as, seeing the emulator embedding the stories of patients, even users not being in healthcare domain seemed to understand what stories and venues were. This had a relevant impact not only on the itsme emulator per se, as we will see later, but also on the services supporting the adopters of itsme we planned.

Some of the people who tested the emulator considered it, per se, as an interesting platform for Web services: when a service extends in time, they claimed its users live it as a story, and giving them a place where they can manage it efficiently may be an added value. These unexpected comments convinced us to start the development of a platform for Web services, called itsociety, whose prototype had positive evaluations.

It was a strong confirmation that our concept was opening a new perspective on the use of ICT, not only a new interface for personal computers.

## 10.8 Continuity and Innovation

But the problems users encountered with the itsme emulator brought to our attention a problem that we had not evaluated adequately. We were aware that, in order to convince users of existing personal computers to move to machine equipped with a radically new operating system, we needed to reduce the cost of this transition. For this reason, we had already planned to develop, together with itsme, a service supporting the transfer of their previous content to itsme. This service will grant that all transferred files will find applications for managing them on itsme (take into account that our system is Linux based) and will allow users to create their first venues, if they want. But this is not enough. The question is more serious. Itsme must be a system people can use without extra effort: this means that, when they access it, they must immediately understand what they have in front, what they can do, and, finally, how they can do what they want, even if what they want is doing as before! It seems a paradoxical requirement: itsme must be, at the same time, as any existing system and radically different from all of them. We can formulate this problem in terms of continuity and innovation: the stronger the innovation of your system, the stricter its continuity with respect to existing systems! We try to face this challenge offering to users a front end where they can behave like on a desktop, but they can also adopt venues, if they want and if they see that they offer

better services. For this reason, what itsme does for all its users is creating a unique space where all objects (of any type) and resources (from tools to people and to information resources) are stored. Objects and resources related with a story can be moved in a venue, when the user decides it (and from that moment, all what is related with that venue goes directly to it). We are convinced that users, creating and using one venue, will discover its effectiveness and appreciate its services, so that they will move to venues more of their stories and of their content, but we avoid to force them toward venues: in some sense, itsme is an open arena where the user can choose among the desktop and the venues and any mix of the two. This choice characterizes itsme as an open tool where different users may adopt different ways of organizing their personal computers: they can choose when creating a venue (at the very beginning of a story or when it has consolidated), how many venues they create, conversely how much they leave in the limbo, how large is the span of a venue, etc., and they can change any of their choice, whenever they want.

A second choice that we did toward making migration from existing personal computers to itsme as smooth and easy as possible has been to use on itsme all applications available on Linux. This choice has also a technical justification: our software layers on top of Linux do not impact the applications, so that we need only to wrap applications in our middleware, without modifying them, containing our effort at software development, but have been reinforced and confirmed at the user side. The user who moves from his/her machine to itsme is not requested to learn to use new software packages, but he/she can continue to use the software he/she was using before or something quite similar to it (we must take into account that, in any case, if he/she must move from Mac OS or Windows to Linux, he/she finds an application only that is similar to some of the most popular ones he/she was using before).

It can also be interesting to recall that this choice convinced us to restart the design of the prototype. The latter was almost ready, when we thought that the choice of having light applications running on it, instead of some standard ones, would be unable to offer a clear experience of working with itsme. We decided, therefore, to restart our development with both a fully serviced suite of productivity tools and an e-mail client: in this way, interacting with our prototype will be a realistic example of what itsme will be and the user will be in the position to fully understand how much we were able to grant a good combination of continuity and innovation.

## 10.9 Conclusion: The Next Challenges

The prototype is currently ready, and we are now preparing ourselves to the next steps. These will be devoted to make the important move from the prototype to the system, even if in beta version. This requires evaluating the prototype to get clear guidelines for improving its usability, developing the services needed to support the transition to it by users of current personal computers, and, finally, defining

a strategy for going to the market. Going to the market requires finding investors and/or partners to package our system in such a way that its potential users (in particular a first small group of them open to innovation) will be convinced to adopt it: we will live even conversations of this type as a step of our design process, because it will be a test-bed, from a particular angle, of the qualities of our system.

We move into this new phase of our project with an open mind and a lot of curiosity and, also, with a reasonable confidence that what we have done up to now has a value, but without any certainty that we will be successful.

We will get back with you after the beta test.

**Acknowledgments** The “stories and venues” metaphor and the itsme workstation project are indebted with many persons, even if the responsibility of what I have written is only mine, and it is not possible to list all of them. Let us recall at least, on the one hand, the itsme team (from the people who participated in the design of the prototype to the business angels that gave us the means for doing it and to the numerous people who expressed interest in what we are doing), who shared with me this unique experience and, on the other, the community of CSCW (computer-supported cooperative work) whose ideas and discussions constituted the basis of our work.

## References

- Agostini, A., De Michelis, G., Grasso, M. A., Prinz, W., & Syri, A. (1996). Contexts, work processes, and workspaces. *Computer Supported Cooperative Work*, 5(3), 223–250.
- Blackler, F., Reed, M., & Whitaker, A. (1993). Editorial introduction: Knowledge workers and contemporary organizations. *Journal of Management Studies*, 30(6), 851–862.
- Bullen, C. V., & Bennett, J. L. (1990). Learning from user experience with groupware. In *CSCW'90: Proceedings of the 1990 ACM conference on computer-supported cooperative work* (pp. 291–302). New York: ACM Press.
- Cabitza, F., Simone, C., & De Michelis, G. (2014). User driven prioritization of features for a prospective inter-personal health record: Perceptions from the Italian context. *Computers in Biology and Medicine*, 59, 202–210.
- De Michelis, G. (2003). The design of interactive applications: A different way. In P. Spirakis, A. Kameas, & S. Nikolettseas (Eds.), *Proceedings of the international workshop on ambient intelligence computing* (pp. 101–114). Athens: CTI Press.
- De Michelis, G., & Grasso, M. A. (1994). Situating conversations within the language/action perspective: The Milan conversation model. In *CSCW'94: Proceedings of the 1994 ACM conference on computer supported cooperative work* (pp. 89–100). New York: ACM Press.
- De Michelis, G., Loregian, M., & Moderini, C. (2009). itsme: Interaction design innovating workstations. *Knowledge, Technology & Policy*, 22(1), 71–78.
- Eppler, M. J., & Mengis, J. (2006). The concept of information overload: A review of literature from organization science, accounting, marketing, MIS, and related disciplines. *The Information Society*, 20(5), 325–344.
- Freeman, E., & Gelernter, D. (2007). Beyond lifestreams: The inevitable demise of the desktop metaphor. In V. Kaptelinin & M. Czerwinski (Eds.), *Beyond the Desktop Metaphor* (pp. 19–48).
- Isaacson, W. (2011). *Steve Jobs: The exclusive biography*. New York: Simon and Schuster.
- Kaptelinin, V., & Czerwinski, M. (Eds.). (2007). *Beyond the desktop metaphor*. Cambridge, MA: The MIT Press.
- Kawell, L., Jr., Beckhardt, S., Halvorsen, T., Ozzie, R., & Greif, I. (1988). Replicated document management in a group communication system. In *CSCW '88: Proceedings of the 1988 ACM conference on computer-supported cooperative work* (pp. 395–404). New York: ACM Press.
- Kay, A. (1977). Microelectronics and the personal computer. *Scientific American*, 237(3), 230–244.

- Mosco, V., & McKercher, C. (2007). Introduction: Theorizing knowledge labor and the information society. In *Knowledge workers in the information society* (pp. vii–xxiv). Lanham: Lexington Books.
- Reder, S., & Schwab, R. G. (1988). The communicative economy of the workgroup: Multi-channel genres of communication. In *CSCW'88: Proceedings of the 1988 ACM conference on computer-supported cooperative work* (pp. 354–368). New York: ACM Press.
- Reder, S., & Schwab, R. G. (1990). The temporal structure of cooperative activity. In *CSCW'90: Proceedings of the 1990 ACM conference on computer-supported cooperative work* (pp. 303–316). New York: ACM Press.
- Sanjay Kairam, S., Brzozowski, M., Huffaker, D., & Chi, E. (2012). Talking in circles: Selective sharing in Google+. In *CHI'12, Proceedings of the 2012 SIGCHI conference on human factors in computing systems* (pp. 1065–1074). New York: ACM Press.
- Schmidt, K. (2002). The problem with 'awareness': Introductory remarks on 'Awareness in CSCW'. *Computer Supported Cooperative Work*, 11(3–4), 285–298.
- Siegler, M. G. (2009). Google wave drips with ambition. A new communication platform for a new web. *Tech Crunch*. <http://techcrunch.com/2009/05/28/google-wave-drips-with-ambition-can-it-fulfill-googles-grand-web-vision/>
- Suchman, L. A. (1993). Do categories have politics? *Computer Supported Cooperative Work*, 2(3), 177–190.
- Suchman, L. A. (2007). *Human – Machine reconfigurations – Plans and situated actions* (2nd ed.). New York: Cambridge University Press.
- Winograd, T. (1994). Categories, disciplines, and social coordination. *Computer Supported Cooperative Work*, 2(3), 191–197.
- Winograd, T., & Flores, F. (1986). *Understanding computers and cognition: A new foundation for design*. Wilmington, DC: Intellect Books.