Chapter 5
Situated Computing

Giorgio De Michelis

5.1 Introduction

Some members of EUSSET (European Society for Socially Embedded Technologies), the European professional association dedicated to the development of technological tools and infrastructures that incorporate a human-centred design perspective, presented at one of the workshops accompanying the development of the new R&D programme of the European Union, Horizon, a position paper (Bannon et al. 2012) where situated computing is proposed as a new paradigm engaged with design and development of technologies from a perspective of evolving social practices.

The position paper is not the right place where to fully develop the argumentation supporting a new user-oriented technological paradigm. Moreover, the position paper is well grounded on the European research on CSCW and related topics, regarding the characterization of the human/social side of the issue, but is opening a new front at the technology level that is not typical of a human-centred design community.

As a coauthor of the above-mentioned position paper, I am willing to stress the novelty of this move inside technology, offering some arguments to a discussion inside and outside EUSSET.

The paper grounds 'situated computing' on the change that has happened on how ICT technology is used, today. Then it recalls the situated action paradigm, as the viewpoint that has been able to recognize and conceptualize the above change. Situated computing is presented as the natural counterpart of situated action and it is surveyed how the concept has been formulated in the last ten years. Finally,

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

V. Wulf et al. (eds.), *Designing Socially Embedded Technologies in the Real-World*,
Computer Supported Cooperative Work, DOI 10.1007/978-1-4471-6720-4_5

I propose my view on situated computing and I introduce some features situated computing systems should have.

5.2 A Change in Use

To shorten our historical account, let us go back to the decade between mid-1980s and mid-1990s: in 1984, Apple launched Macintosh, and in 1992 SAP presented its Sap R/3 system, later named SAP ERP.

Why are these dates important? Because Macintosh, at the personal computing level, and Sap R/3, at the organizational computing level, fix, from the user interaction viewpoint, the standards that characterize, in their domains, all the systems that are proposed to the market up to current days. It is not by chance in fact that all operating systems for personal computers resemble each other, sharing the desktop metaphor, invented by Alan Kay at Xerox PARC in the late 1970s (Kay 1977) and made popular by Macintosh, and that the same is true for ERPs, adopted by the majority of medium and large companies all around the world (Kumar and Hillergersberg 2000).

Despite the evolution that both operating systems for workstations and ERPs have had in these 20 and more years, due to the combined effect of continuously growing memories, better communication channels, the web, etc., both remain, structurally and from human interaction viewpoint, the same.

When they conquer the markets, both are in essence multifunction devices supporting users in a growing variety of tasks. Their success depends on their capability to solve the problems affecting the diffusion of ICT within work environments: the simple and highly usable interface of Macintosh and imitators allows everyone to use it, and the strong integration of ERP systems gives to enterprises the possibility of planning, managing, and controlling their operations in a smooth way.

Until people use them for doing their tasks, in fact, their support is effective, even if the number of tasks and the amount of stored data and/or documents grow. Accessing data and files is considered one function among the others and what matters is being able to find a document (in the workstations) and to process data (in the ERPs).

But changes of ICT have induced unexpected changes in the way its applications are used: some users, in particular, have slowly but irreducibly changed the role they attribute to ICT. Let us look closely at those for whom reading and writing documents, searching and elaborating information, and collecting, interpreting and calculating data constitute the texture connecting all their activities. They have in the digital world (directly in the workstation or in the web or, finally, in the information system of the organizations with which they are collaborating) almost all what they need both in terms of content and functions (from an organizational science viewpoint, they are called knowledge workers; see, e.g., Blackler et al. 1993; Drucker 1999; Mosco and McKercher 2007). These people have progressively ceased to approach their PC (and what there is behind it) when they have to perform a task, but rather, always more frequently, they have it always on, and whatever they are doing, they switch to it to find and/or do what may help them to act and interact effectively. In other words, when the number of tasks they are executing with the support of ICT has become so large and, mainly, when the amount of their digital data and documents has become so big that they are almost always interacting with a digital device, it happens that ICT systems progressively change, from tools for executing tasks to relational extensions of their capability to act and interact. The workstation (or, in other circumstances, the tablet and/or the smartphone) is always on, because it plays a crucial role, in any situation, allowing the user to act and/or interact effectively: sometimes it is an extension of his/her memory, sometimes it is a powerful communication channel, and sometimes it is necessary for executing tasks like writing and/or calculating and the like. But what is required is that, in any situation, contents, communication channels, and productivity tools are filtered so that all and only what may be useful is accessible.

While, as tools for executing tasks, current ICT-based systems are user-friendly and effective, as companions to act and interact effectively, they appear always more cumbersome and inefficient (the literature on this issue is rich; see, e.g., Kaptelinin and Czerwinski 2007; Eppler and Mengis 2006; Monsell et al. 2000; Oulasvirta 2008; Yeung et al. 2006). For accompanying users in any situation they may encounter during their (working) day, in fact, it is not sufficient that tools are user-friendly; what is also and mainly needed is that, in any moment, all and only what the user needs for acting and interacting (data, documents, messages, information resources, tools, communication channels, people) is ready at his/her hands. But this is not the case, because both workstations and ERPs are multitasking systems and are not capable to provide help depending on the situation of the user.

Let me repeat one thing, to avoid misunderstandings: I am not critiquing the design of the Macintosh and/or of SAP R/3; rather, I am only claiming that they were designed for a time when machines and the way of using them were different and that they, today, have become inadequate. The desktop, emulated by operating systems for personal computers and workstations, is not well designed neither from the point of view of keeping all the things a user needs altogether (it is well known that, while documents created by the user are in the folders of the file system where he/she has stored them, messages are inside the mail system, their attachments are in a special folder of the file system and documents from the web are memorized as URLs in a list) nor from the viewpoint of collecting them in separate spaces. The problem of putting together things related with the same user issue is left to his/her goodwill of creating and updating dedicated folders fighting with the obstacles its organization opposes to him/her. In the same vein, ERP systems are highly integrated monolithic systems that are quite efficient at integrating data with respect to routine or expected tasks, but, conversely, they are quite rigid with respect to free access and unexpected processing.

5.3 Situatedness

The issue, here, is understanding how things are correlated with respect to users, what should be ready at their hands whenever they need to interact with their system. Research on CSCW and related topics in the last 30 years has deeply investigated it, bringing forth several hints on human practice in different work contexts and situations. Situated action (Suchman 1987), language action perspective (Winograd and Flores 1986), and embodied interaction (Dourish 2001) are some of the headings that have been formulated in the effort to characterize what is constitutive of human practice and relevant for the design of ICT-based systems supporting it. Altogether, they underline that human practice is intrinsically social, that it is situated, and that what people say is strictly and bidirectionally linked with what they do.

These three hints on human practice recall that the effectiveness of human beings strongly depends on the awareness they have of their situation, i.e. of the context where they are situated. From what we have said above, a question emerges with big evidence: in the frame of the discourse that we are carrying on, what is the context that people should be aware of? The answer is not as immediate as anyone could think at first moment: there are, in fact, different dimensions of the context where a person acts and interacts (see e.g. Kishore et al. 2004).

First, we can assert that it is not the spatial context: the place where a person is (with the various things and people populating it) strongly affects what she can do and what she cannot do and being aware of it is important, but we cannot say that it reflects the social dimension of human experience.

Second, we can also assert that it is not the temporal context: in different days, at different times of the same day as well as of any day, what she can do and not do changes, as well as what she should and should not do, but social experience evolves in a way that goes beyond strict temporality.

Beyond spatial and temporal contexts, there is a 'social context' that, in some sense, includes both of them: with whom is the person engaged while acting and interacting? What are they doing together and what is the aim of their interactions? What have they already done and what are they mutually committed to do? Whatever a person is doing, this is part of an experience she is living with some other people, with whom she exchanges documents and other things, she shares information and knowledge, and she has mutual commitments. All what participants do during a social experience intertwines language and action, so that, at the same time, things are created, imported, and/or modified and knowledge is created and shared.

The thread of events constituting a social experience creates also the language (game; this implicit reference to Wittgenstein (1953) is not casual) and the knowledge shared by its participants, and for this reason, being aware of the context in which she is acting and interacting is, for a person, necessary in order to be effective in it. We call this thread a 'story', to underline its sense-making role for its participants. All events of what we have called a story are, naturally, situated in space and their thread develops in time: on the one hand, a story, taking place in a space, where its actors live their common experience, transforms that space in its place (Harrison and Dourish 1996); on the other hand, any story has a duration, during which it evolves reacting to the events involving its actors and to the mutual commitments they have established.

Social context is of paramount importance, in particular, for those people who are involved in many different stories, because whichever is the story she is acting and interacting and what is happening in others is, both, disturbing (creating noise and confusion in it) and enriching (opening it to new knowledge) it.

For our target users, i.e. the knowledge workers, whose work is woven of what they read and write, the issue is not if they are using the word processor, the spreadsheet or the mail, but which is the story in which they are engaged so that they can act effectively in it.

5.4 Situated Computing

Systems supporting human practice should therefore be able to improve context awareness of their users, so that they can act and interact effectively in any situation of their life. Researchers in CSCW and related areas have become aware of this fact from many years, as the emergence of a new heading 'situated computing' shows with great evidence.

The term 'situated computing', in fact, is not new and it may be useful to survey its history. For what I know, it is in the second half of the 1990s (1997) that it was used for the first time, by three researchers of the HP Laboratories, R. Hull, P. Neaves, and J. Bedford-Roberts, in a paper they presented at the First International Symposium on Wearable Computers (ISWC '97): 'Towards situated computing'. In the abstract, they wrote: 'Situated computing concerns the ability of computing devices to detect, interpret and respond to aspects of the user's local environment' (Hull et al. 1997). Using the terminology we introduced in the previous section, the authors make reference to spatial context. From 1997 to present days, several other authors have revived the term proposing their view on it.

Let us survey some of the contributions appearing in the literature and some of the initiatives launched under this heading.

In 2001, Masahito Hirakawa and K. Priyantha Hewagamage published 'Situated computing: A paradigm for the mobile user-interaction with multimedia sources' in the *Annals of Software Engineering*. In the abstract they wrote: 'Situated computing is a new paradigm for mobile computer users based on their physical context and activities carried out as a part of their working business. It provides the mechanism to have a mobile computer as a utility to satisfy the user's real world requirements as well as an infrastructure for the situated interaction using applications' (Hirakawa and Hewagamage 2001). Here the authors are again narrowing their use of the term for mobile systems, focusing on spatial context.

One year later, Kevin L. Mills and Jean Scholtz published ‘Situated computing: The next frontier for HCI research’ in a book edited by J. M. Carroll, *HCI in the New Millennium*. In the paper it is written: ‘An impressionist painting emerges of nomadic workers with collections of small, specialized devices roaming among islands of wireless connectivity within a global sea of wired networks. Each wireless island defines a context of available services, embedded devices, and task-specific information. As nomadic workers roam the landscape the context in which they are working continuously changes. As workers move onto wireless islands of connectivity, their context is merged with the context of the island to automatically compose a computational environment to support their needs. At other times, when not connected, an array of portable devices provides each nomad with a local context for computing. This painting, which relies heavily on Weiser’s (1991) concept of ubiquitous computing and on Suchman’s (1987), notion of situated computing, suggests a future where information and people connect directly and work together across a range of contexts’. (Mills and Scholtz 2001). Even if here ‘situated computing’ assumes a visionary character for becoming the label of what could be the future of computing, it has to be remarked that while the relationship between space and mobile devices is well developed, the same cannot be said about the ‘user’s context’.

It goes in the same direction when, in 2005, John S. Gero presented ‘Virtual Environments Using Situated Computing Can Change What We Design’ at Virtual Concept, a conference held in Biarritz (France). This paper considers situated computing as a new design paradigm. Its abstract says: ‘This paper presents the foundational concepts of situated computing: first-person interaction, constructive memory and situations. It then describes two classes of situated design that differ from other forms of designing: situated interaction design and situated artifact design’ (Gero 2005). The social nature of situated action is not considered, and in some sense we can consider situatedness as a combination of individual memory and spatial context.

In 2009, the call for the 15th International Conference on Distributed Multimedia Systems claims: ‘DMS conference is an international conference series, which covers a wide spectrum of paper presentations, technical discussions and demonstrations in the fields of distributed multimedia computing. … The main themes of the DMS2009 conference are: network and systems, emergency management and security, *situated computing*, multimedia software engineering, and multimedia information retrieval, mining and fusion’; the term has become a label for describing one of the themes of a conference focusing on distributed multimedia computing.

In 2011, Inderscience started publishing a new *International Journal of Space-Based and Situated Computing*. Its aim is extending ‘the pervasive computing vision of everyday objects communicating and collaborating to provide intelligent and context-aware information and services to users in larger geographical spaces. The ultimate goal is to build context-aware global smart space and location-based service applications that integrate information from independent systems (such as sensors, actuators or mobile information systems), which autonomously and securely support human activities. *IJSSC* provides a fully refereed international forum for publishing the latest research into space-based and situated computing'. The term has found its place in the scientific community, but again, its focus has been narrowed to spatial contexts.

But, from the very beginning, there are contributions that go in a direction echoing in a stricter sense than what we have discussed in the previous section.

In 1999, A. V. Gershman, J. F. McCarthy, and A. E. Fano presented 'Situated Computing: Bridging the Gap between Intention and Action' at the 3rd International Symposium on Wearable Computers. In the abstract, they wrote: 'Situated computing represents a new class of computing applications that bridges the gap between people's intentions and the actions they can take to achieve those intentions. These applications are contextually embedded in real-world situations, and are enabled by the proliferation of new kinds of computing devices, expanding communication capabilities and new kinds of digital content. Three types of discontinuities give rise to intention/action gaps and provide opportunities for situated computing applications: physical discontinuities, information discontinuities and awareness discontinuities' (Gershman et al. 1999). Here the authors make reference to a broader view of context, where its social dimension is taken into account.

In 2000, Michel Beaudouin-Lafon and Wendy E. Mackay organize at CHI2000 a workshop on situated computing. In the presentation of the workshop, it is written: 'The term *situated computing* describes socio-technical systems in which situations of use and context play a central role in the use of computers. Since most computing is arguably situated computing, we need to reflect on our current understanding of context, establish a common language for discussion and define processes for developing *systems-in-use*' (Beaudouin-Lafon and Mackay 2000). Among the contributors to the workshop, there is Paul Dourish (A Foundational Framework for Situated Computing; 2000) who, in his position paper, wrote: 'One starting point for this exploration is a conundrum which was, interestingly, raised for me by the call for this workshop. The call coins the term "situated computing" to refer to the set of technologies and usage experiences that make up the burgeoning area of contextually informed system design. The term I use myself is "Embodied Interaction" (for reasons that will become clear. However, I think "situated computing" is an excellent term, because it captures two distinct elements of the area. First, it captures its technological foundations, and the relationship to other, related technological explorations such as the Ubiquitous Computing work spearheaded at PARC in the early 1990s. Weiser (1991) set out a vision of a world in which technology supported us more intimately by retreating into the background, one in which the world around us was imbued with computational power that could be called upon intrinsically as part of everyday activity. At the same time, the word "situated" evokes the "situated action" perspective that has played a dominant role in the sociological foundations of Computer-Supported Cooperative Work. Suchman (1987), drawing on the ethno-methodology of Harold Garfinkel [5], radically revised cognitivist accounts of natural activity to turn attention to the improvised and contingent nature of the sequential organization of activity – its situated character'.

With this workshop, the proponents underline that situated computing should not focus on a particular class of systems, since it refers to a feature that is relevant for a very large variety of the ICT-based systems already in use, but, maybe, because the debate is still restricted in a small group of specialists within the CSCW/HCI research community, it is not clear which situated computing systems do the participants have in mind.

Concluding this historical survey, it may be useful to recall that ‘situated computing’ with the attention it calls for the coupling between situatedness and ICT-based systems has given rise to the term ‘situated software’ (Balasubramaniam et al. 2008) and, more recently, has been used by Carlo Ghezzi and co-workers for characterizing the change in perspective they propose in software engineering, taking into account that most software development is contextualized, since it aims to modify existing running systems (Salvaneschi et al. 2012).

I do not pretend that my survey is complete, but, I think, it suffices for showing that situated computing has become a popular header and that it is used with different meanings, moving from the identifier of the emerging class of mobile location-aware computing systems to the label for a new paradigm for the design of information systems and services. Even from this radical viewpoint, however, it has not yet opened its eyes towards the systems already in use, neither to evaluate them from a viewpoint going beyond task-oriented use nor to figure out how they can be redesigned.

5.5 A New Definition of Situated Computing

As said above, situatedness recalls the existence of contexts, and designing systems for it requires to characterize the latter in a way suitable for understanding what computing systems may do to help user situation in them. Contexts, we have recalled, are, intrinsically, multidimensional: they are spatial, since in any moment a person is situated in a portion of space that can be, sometimes, a place she inhabits; they are temporal, where we intend time with all its facets – the absolute one, when we consider the current date and hour, or the relative one, when we consider the current part of the day (morning, afternoon, etc.); they are social, when we consider the social relations in which our actions and interactions are immersed. Making a drastic schematization, we can claim that the social dimension embodies the other ones, since social relations develop in space and time and play a major role in configuring the sense-making vector of any action or interaction. What a person is doing gets, in fact, its sense from the actor network (Latour 2005) which she is part of, from its past events and from its events and mutual commitments scheduled in the future. Summarizing shortly, a social context is generated by a thread of past and future events involving an actor network within which its participants create their common place and knowledge for sense making and effective action and interaction. Its complexity has two orthogonal dimensions: on the one hand, sharing an experience is never definitive – even if and when we live together, we cannot share what we feel; on the other hand, we are generally engaged in several different social experiences: this gives us the capability to enrich any experience in which we participate and, conversely, may distract us from contributing effectively to any one of them.

Situated computing has to do, therefore, with supporting people in the threads in which they participate. This may require the design of new applications but, mainly, it needs a strong help from the applications a person is already using: first, the workstation she uses as the principal terminal for organizing her augmented workplace so that she is kept aware of her threads; second, the information resources of the net and the information produced within the organizations which she is part of that may be necessary for performing in those threads. But, as they are now, neither the operating system for workstations is capable to support user participation in her threads nor information resources (both in the net and in the information systems of the organizations) are offering effective support; instead, they are making always more complex for users situating themselves in the context where they are operating.

Situated computing calls for systems which are designed to take into account the situatedness of human action. It challenges ICT scholars and professionals to redesign the most diffused ICT applications, like operating systems of personal computers and other mobile devices, on the one hand, ERPs and other organizational computing systems, on the other.

5.6 Some Hints on Possible Situated Computing Systems

The generic definition I gave above may leave many readers unsatisfied. Therefore, I add to it some preliminary hints on how situated computing systems can be designed and how I am doing some work in this line. I underline the adjective ‘preliminary’, I have used in the lines above, because I cannot make any strong assertion on the subject and I do not think that it would be serious to make strong assertions on it. Being specific with respect to ‘situated computing’ is not possible because we are speaking about not yet existing systems and we do not have any real experience with them, so that we cannot evaluate the qualities of the software inspired by it, without people experimenting it.

I will dedicate two separate subsections to (1) systems for end users (front ends) and (2) big data repositories and systems for managing organizational information (back ends), like in any client–server architecture. The front end needs to adhere directly to the needs and desires of the user, while back ends need to be open to what front ends may require for serving their users.

At front end side, it is necessary that systems (from personal computer operating systems to web-based services for mentioning the two most important classes of systems that are characterized by their front ends) are designed so that the context of usage is accessible without noise and confusion. This requires that front ends are designed keeping together the events constituting threads. We have designed both an operating system for personal computers (De Michelis et al. 2009; De Michelis 2015) and a platform for web services (De Michelis 2014) on the basis of a new metaphor, called ‘stories and venues’, considering the life of human beings as the intertwining of several different stories and considering for each story the venue where participants find all that is relevant in it. Stories, as sets of threads, are not objective phenomena; rather, they are quite subjective – different people may group in a story different threads – but they are not arbitrary, because threads are, in our approach, the new atomic elements of human experience. How can a story be captured/reflected in a digital application? Organizing the user workspace so that she can access, for each story, to all and only what characterizes it and she can move among her stories. The apparent contrast between *multiplicity* (of stories) and their *openness* can be solved putting at their boundaries the resources needed to grant *continuity* (Brown and Duguid 1994; De Michelis 1998, 2003).

At back end side, the question is quite different: the problem is making any system supporting services or containing data and/or information that users may need to be as open and accessible as possible. Disregarding, here, web-based systems making public large amount of information, for whom the perspective of open linked data promises the needed developments, let us to discuss here, briefly, systems containing the information of organizational systems like ERPs, because, as they are today, they are quite far from what situatedness requires (Dörner et al. 2009).

ERPs and the likes owe their large popularity and diffusion to the fact that they were able to integrate the different information generated by an organization (logistics and accounting, marketing and production, etc.) creating a unique database reflecting a well-defined organizational model that serves all the functions of the organization. This choice has determined a strong standardization of the architecture of information systems (whose efficiency in the routinary tasks is quite high), as well as of the ways to build them (building an ERP is a well-defined task that can be completed in less than 1 year), but, conversely, has made any information processing that is not defined in accordance with the standards characterizing the system difficult (generally a 'mining' activity is needed in this case). The growing relevance that business intelligence and strategic planning have within organization tells us that unforeseen processing of organizational information is becoming a frequent and non-exceptional need: how can we couple the efficiency of existing ERPs with flexibility?

Can ERP systems be redesigned so that innovation and changes to information processing can be possible despite the strong integration of applications they imply?

As a blueprint for this objective, I shortly indicate what follows. First, we can redesign ERP systems as modular systems made of small modules separating functions and data (a similar proposal can be found in Dorner et al. 2009). This means that users will build their system in a bricolage-like style (Ciborra 1999) selecting their components in a large library of modules and linking them in accordance with the business intelligence governing the organization. It has to be underlined that designing the new generation of ERP systems sketched above is possible today, thanks to some ICT technology that has appeared today.

First, cloud computing (Armbrust et al. 2010), among other potential advantages deriving from its capability to support a pay per use policy, offers a strongly homogenous platform simplifying the construction of modular systems; second, mashup technology allows a flexible merging of several APIs needed for interacting with different organizational systems and websites in the front end of user workstations; third, ontologies, supporting an effective tagging of software objects, allow to substitute the vertical integration typical of traditional ERPs with the light coupling you can get through metadata.

It may be useful to underline, here, that modularity has been for long time a concept that scholars and practitioners considered capable to increase the quality of systems, but it has been obstructed, up to now, for its difficult feasibility in heterogeneous computing environments. Cloud computing, with its homogeneous hardware and software platform, together with mashup and ontologies, dissolves the obstacles modular systems have encountered up to now.

5.7 Conclusion

All the new systems we can design and build along the 'situated computing' perspective constitute a great challenge for the community of CSCW and related area researchers, since all of them will strongly influence our engagement with the development of ICT technology.

It is a challenge that they can accept, because the features characterizing 'situated computing' systems emerge from an interaction design (Telier 2011) approach. Their further characterization is not possible without the adoption of the same approach that those disciplines dedicate to new and emergent applications like Web 2.0, collaboration systems, and the likes, for mature systems that have conquered a solid and permanent position in organizations.

But the solidity and duration of existing ICT technologies on the desktops and in the inner parts of organizations tells us that innovation in that field requires a great cure in managing the transition from existing to new systems. The migration from an ERP to its modular replacement is a complex process, both at the human (organizational) and technological levels, requiring that new systems are designed for supporting this migration.

Even more difficult is the transition from operating systems based on the desktop metaphor to the new ones that can be designed along the situated computing perspective, since, here, we deal with human behaviour and expectations. It is well known that innovative systems may fail, because users refuse to pay the price to abandon their system (which is transparent to them, even when ineffective) for adopting the new one (which requires an extra cognitive effort for being used). This means that the design of a new operating system for workstation must couple its innovativeness with the highest degree of continuity with the systems it wants to substitute. It seems a strange paradox, but it indicates the critical quality supporting the adoption of innovation in areas where technology is already present.

Acknowledgments As recalled in Introduction, this paper is an extension of my contribution to the position paper Liam Bannon, Pernille Bjørn, Fabio Paternó, Dave Randall, Kjeld Schmidt, Ina

Wagner, and Volker Wulf and myself presented to the 2nd FIA Research Roadmap Workshop 'Looking to the Horizon – Future Internet Assembly Research Roadmap for Horizon 2020' (Bannon et al. 2012). Before that occasion I had several occasions to discuss situated computing in Italy and abroad. It has been quite relevant for me the panel on this theme at CTS2011, with Steve Benford, John Carroll, Elizabeth Churchill, and Prasun Dewan, where my first formulation of a situated computing manifesto has been deeply and constructively critiqued. Federico Cabitza and Carla Simone are currently engaged with me in a book project on situated computing where my views are enlarging under the inspiration of their viewpoints, theorizations, and experiences. I thank also the reviewers and the editors of this book for the valuable comments on its first version that helped me to try to improve its readability and effectiveness.

References

Armbrust, M., Fox, A., Griffith, R., Joseph, A. D., Katz, R., Konwinski, A., Lee, G., et al. (2010). A view of cloud computing. *Communications of the ACM, 53*(4), 50–58.

Balasubramaniam, S., Lewis, G. A., Simanta, S., & Smith, D. B. (2008). Situated software: Concepts, motivation, technology, and the future. *IEEE Software, 25*(6), 50–55.

Bannon, L., Bjørn, P., De Michelis, G., Paternó, F., Randall, D., Schmidt, K., Wagner, I., & Wulf, V. (2012). *Building a socially embedded future internet*. Paper presented at 2nd FIA research roadmap workshop – Looking to the horizon, future internet assembly research roadmap for horizon 2020, 2012. Germany: University of Siegen.

Beaudouin-Lafon, M., & Mackay, W. E., (Organizers). (2000). Workshop on research directions in situated computing, at the *ACM conference on human factors in computing systems CHI 2000*, Position papers. Netherlands: The Hague.

Blackler, F., Reed, M., & Whitaker, A. (1993). Editorial introduction: Knowledge workers and contemporary organizations. *Journal of Management Studies, 30*(6), 851–862.

Brown, J. S., & Duguid, P. (1994). Borderline resources: Social and material aspects of design. *Human–Computer Interaction, 9*(1), 3–36.

Ciborra, C. (1999). A theory of information systems based on improvisation. In W. Currie & R. Galliers (Eds.), *Rethinking management information systems: An interdisciplinary perspective* (pp. 136–155). Oxford: Oxford University Press.

De Michelis, G. (1998). *Aperto, molteplice, continuo*. Milano: Dunod Italia.

De Michelis, G. (2003). The Swiss Pattada: Designing the ultimate tool, (with original drawings by Marco Susani). *Interactions, 10*(3), 44–53.

De Michelis, G. (2014). *Open social services* (manuscript), available from the author.

De Michelis, G. (2015). Interaction design at itsme. In V. Wulf (Ed.), *Designing socially embedded technologies in the real-world* (pp. 193–215). London: Springer.

De Michelis, G., Loregian, M., & Moderini, C. (2009). Itsme: Interaction design innovating workstations. *Knowledge, Technology & Policy, 22*, 71–78.

Dorner, C., Draxler, S., Pipek, V., & Wulf, V. (2009). End-users at the bazaar: Designing next generation enterprise resource planning systems. *IEEE Software, 26*(5), 45–51.

Dourish, P. (2000). *A foundational framework for situated computing*. Mackay: Beaudouin-Lafon.

Dourish, P. (2001). *Where the action is: The foundations of embodied interaction*. Cambridge, MA: MIT Press.

Drucker, P. F. (1999). Knowledge-worker productivity: The biggest challenge. *California Management Review, 41*(2), 79–94.

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.

Gero, J. (2005). Virtual environments using situated computing. In X. Fischer & D. Coutellier (Eds.), *Proceedings of virtual concept 2005*.

Gershman, A. V., Mccarthy, J., & Fano, A. (1999). Situated computing: Bridging the gap between intention and action. In *Proceedings of the 3rd international symposium on wearable computers* (pp. 3–9). San Francisco: IEEE.

Harrison, S., Dourish. P. (1996). Re-place-ing space: The role of place and space in collaborative systems. In: *Proceedings of the 1996 ACM conference on computer supported cooperative work* (pp. 67–76). New York: ACM Press.

Hirakawa, M., & Priyantha Hewagamage, M. (2001). Situated computing: A paradigm for the mobile user-interaction with multimedia sources. *Annals of Software Engineering, 12*(1), 213–239.

Hull, R., Neaves, P., & Bedford-Roberts, J. (1997). Towards situated computing. In *First international symposium on wearable computers (ISWC '97).* Cambridge, MA: IEEE.

Kaptelinin, V., & Czerwinski, M. (Eds.). (2007). *Beyond the desktop metaphor*. Cambridge, MA: MIT Press.

Kay, A. (1977). Microelectronics and the personal computer. *Scientific American, 237*(3), 230–244.

Kishore, R., Sharman, R., & Ramesh, R. (2004). Computational ontologies and information systems: I. Foundations. *Communications of the Association for Information Systems, 14*, 158–183.

Kumar, K., & Hillergersberg, J. (2000). ERP experiences and evolution. *Communications of the ACM, 43*(4), 23–26.

Latour, B. (2005). *Reassembling the social – An introduction to actor-network-theory*. Oxford: Oxford University Press.

Mills, K. L., & Scholtz, J. (2001). Situated computing: The next frontier for HCI research. In J. M. Carroll (Ed.), *HCI in the new millennium*. New York: Addison Wesley.

Monsell, S., Yeung, N., & Azuma, R. (2000). Reconfiguration of task-set: Is it easier to switch to the weaker task? *Psychological Research, 63*(3-4), 250–264.

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.

Oulasvirta, A. (2008). Feature when users "do" the ubicomp. *Interactions, 15*(2), 6–9.

Salvaneschi, G., Ghezzi, C., & Pradella, M. (2012). Context-oriented programming: A software engineering perspective. *Journal of Systems and Software, 85*(8), 1801–1817.

Suchman, L. (1987). *Plans and situated actions*. New York: Cambridge University Press.

Telier, A. (Binder, T., De Michelis, G., Ehn, P., Jacucci, G., Linde, P., Wagner, I.). (2011). *Design things*. Cambridge, MA: MIT Press.

Weiser, M. (1991). The computer for the 21st century. *Scientific American, 251*(11), 94–104.

Winograd, T., & Flores, C. F. (1986). *Understanding computers and cognition. A new foundation for design*. Wilmington: Intellect Books.

Wittgenstein, L. (1953). *Philosophical investigations*. Oxford: Blackwell.

Yeung, N., Nystrom, L. E., Aronson, J. A., & Cohen, J. D. (2006). Between-task competition and cognitive control in task switching. *The Journal of Neuroscience, 26*(5), 1429–1438.