The Situation Lens: A Metaphor for Personal Task Management on Mobile Devices

Augusto Celentano, Stefano Faralli and Fabio Pittarello
Università Ca' Foscari Venezia, Dipartimento di Informatica
Via Torino 155, 30172 Mestre (VE), Italy
{auce,faralli,pitt}@dsi.unive.it

Received 26 May 2009; Accepted 25 August 2009

In this paper we discuss personal data management with mobile devices, an activity requiring the composition of services offered by standard suites of applications. We propose a data model and an interface model that allows users to define activities, tasks and services, to navigate among them according to the evolution of the personal situation as perceived and interpreted by the users themselves. The interface model acts as a lens exploring the situation, zooming into the details, covering different areas of the personal data, supporting the user in the role of a composer of personal services.

Categories and Subject Descriptors: H.5 [**Information Interfaces and Presentation**]: User Interfaces

General Terms: Personal Information System, Personal Digital Assistant

Additional Key Words and Phrases: Service Composition, Situated Computing, Data Model

1. INTRODUCTION

In this paper we discuss issues about the execution of tasks related to personal activity with portable devices. Devices like UMPC, PDA, smartphone, etc., offer standard suites of applications for personal data management: an agenda, an address book, todo lists, mailing and messaging services, maps, etc., often independent or loosely coupled, used at different extent according to the personal style and skill of the user. Such applications compel the users to find own way to select and to compose the services needed for managing their tasks, activating sequences of commands related to different applications without the benefit either of an adequate information flow, or of a smooth transition from one application to the other.

In service oriented applications, the problem of finding and composing services is usually approached with an automation perspective: environment sensing, service discovery and task planning are the key terms driving the search for efficient ways to compose black-box services, building applications adaptable to the user needs. The Activity Theory concepts [Bertelsen and Bodker 2003] play an important role in such a view: the services needed to execute an activity are related to a state of the user that can vary only within defined boundaries without influencing the activity content (hence the user goal), but only the way in which the tasks are executed.

This view falls short in many activities related to personal data management, which evolve in space and time as the user proceeds in executing the tasks. The scenario is closer to the one defined by the Situated Cognition approach [Clancey 1997]: actions are driven by the user situation, a complex set of variables related not only to space, time, profile and environment, but also to the user goals, history, emotional status, etc.. The situation evolves as new information is acquired and results of previous tasks are collected. In practice, the user starts by defining a coarse-grain plan that is continuously refined and adapted, evolving according to the many variants in the situation encountered during its execution [Nardi 1996; Suchman 1987].

We propose a model for user task organization and an interface model that links the personal services provided by a Personal Information Management (PIM) suite, allowing users to navigate among them according to the situation evolution as perceived and interpreted by the users themselves, hence not driven by predefined adaptation mechanisms.

While the main features of the model are independent from the specific delivery platform, the model has been validated in a prototype developed on the Windows Mobile platform, because of its diffusion and of the expansibility of its personal information model. Such expansibility, as we shall discuss in Section 5, allows us to reuse the standard set of personal information objects and relations and enhances them as a necessary step for applying our metaphor. Other platforms might be considered in future developments, provided they grant sufficient expansibility; in Section 7 we shall discuss issues related to the Apple iPhone platform.

The PIM in Windows Mobile systems includes several applications that can be viewed as services offered to the user: address book, agenda, messaging, tasks. They are activated as independent programs and refer to information chunks that are independently stored. However, during the execution of an activity, the user switches from a service to another building in his/her mind a map of related data elements. For example, upon receiving a mail message from a colleague, the user might decide to call her instead of replying with a mail message, switching from the mail program to the address book, indexed on the name of the mail sender. In the same way, when reading about the next meeting in agenda, the user could decide to consult a map of the meeting location, e.g., to find a fast way to move there, indexed on the address of the colleague who has organized the meeting.

The developed prototype wraps the PIM, exploiting the available application data in a more complete data model, where additional relations between the informative objects are defined.

The interface model is called *Situation Lens* because, as an optical lens, it allows the user to focus on the different information objects that characterize the situation at different levels of detail, covering different data areas, thus supporting the user in the management of personal tasks.

The ability to change perspective, that allows the user to seamlessly examine information from different points of view, such as people, tasks and locations, is the key features that distinguishes our proposal from other proposals based on the informatin integration but still relying on separate visualizations.

The paper is organized as follows: after briefly reviewing the relevant literature on context adaptation and service composition in Section 2, we discuss the notion of personal service composition and introduce the concept of *Situation Lens* in Section 3 and a model for describing user activities, tasks, services and situation in Section 4. The Situation Lens interface and the service integration model are discussed in Section 5. A prototype implementation is discussed in Section 6. Section 7 discusses the interaction metaphor by comparing it with other models discussed in the literature, while Section 8 draws the conclusions.

2. RELATED WORK

Situated computing is based on the idea that the different parameters that define the user situation may change, requiring a reorganization of the information provided and of the accessible services. It stems from research on context-aware systems and from the psychological studies on situated cognition.

Context-aware computing, and the notion of *context* in general, have been widely extended since the initial focus on location dependent systems [Bennett et al. 1994; Schmidt et al. 1999; Chen and Kotz 2000; Davies et al. 2001]. The definition of context by Dey [2001] is still taken as authoritative, but many authors try to make more explicitly the presence of components related to the user "per se" (see for example [Jameson 2001; Petrelli et al. 2001]). Extending the analysis of the user situation to include the user plans and goals in executing tasks and accessing information requires a view of the context also in cognitive and linguistic domains [Schmidt 2006]. A complete view of context is presented by Bradley and Dunlop [2005] who explore a multidisciplinary approach addressing in a unified view different domains.

Some authors [Dourish 2000] propose the word *situation* as a replacement for the word *context*, because it widens the number of parameters that are significant for the user interaction. In particular, the term *situation* moves from physical, observable features of the user environment to a higher level of context including the user goals, plans, activity and history [Hewagamage and Hirakawa 2000].

Situated Cognition [Clancey 1997; Gero 2003] is a behavior model claiming the dependence of the user actions on the set of environmental, social and psychological variables describing the user situation. It has been studied as opposed to the Activity Theory model [Nardi 1996; Bertelsen and Bodker 2003], based on the definition of action plans to reach anticipated goals. The two action paradigms are not completely disjoint, and relations between them have been studied [Bardram 1997; Lueg and Pfeifer 1997; Seel 2001; Suchman 1987].

Situated computing, as context aware computing, builds on the notion that the different parameters that define a situation (a context) may change, requiring a reorganization of the support to the user, in terms of information provided and of services made accessible. Situated computing is thus strongly related to the concept of information and interaction adaptation, and exploits in mobile computing its full potential [Bradley and Dunlop 2005; Lemlouma and Layaïda 2004]. Context modeling, information adaptation and multimodal interaction are the three keywords denoting the ability to design systems able to perceive the user status, to provide information and services with proper content and presentation, and to offer a user oriented effective interface.

In many domains related to information processing both theories have contributed to the development of models and methods for configuring and adapting personal information systems.

In the domain of document processing the activity theory seems suitable when the user has a precise idea of the meaning and purpose of the documents related to some goal. With the growth of information processing capabilities, and the increase in a variety of devices and media, the understanding of a document cannot be split from the interpretation of the context in which it is delivered to a user. Plans, therefore, should be viewed as general guidelines to build an interactive experience evolving according to the evolution of the user's situation [Boll et al. 1999; Celentano and Pittarello 2007; Celentano 2008; Elouazizi and Bachvarova 2004].

In database systems, namely in the area of personal databases, close to the scope of this paper, an interesting approach to context adaptation is the *Context-ADDICT* project (http://poseidon.elet.polimi.it/ca/), aiming at selecting the part of a database relevant for the context of mobile users, through the dynamic hooking and integration of information sources [Bolchini and Quintarelli 2006; Bolchini et al. 2007; Tanca 2007]. Adaptation is based on an ontological representation of the application domain and of data source contents, tailored on the user context dimensions. Differently from our approach, focused on the user interface level for selecting different perspectives on data, the Context-ADDICT system is targeted to design time, and its scope does not include the user interface and the interaction functions.

In the area of service computing, adaptation has been investigated primarily to find the best set of services to build an application fulfilling a user need. Architectures and strategies for context-aware service composition have been investigated by several authors [Berhe et al. 2005; Costa et al. 2005; Chaari et al. 2007; Strang and Linnhoff-Popien 2003; 2004]. As noted in Section 1, their primary focus is to automate service discovery and composition, while our model puts the user in the center of the service composition process.

User interaction in mobile systems has been investigated under the perspectives of usability and information presentation, supporting the user to seamlessly and naturally navigate through information and services. The small screen size of mobile devices affects the display of overview information more than of detail data. While many studies have investigated the so-called "focus+context" approach to information visualization in large displays, the literature addressing small screens is still limited, and often related to small and large displays coupling for information analysis rather than on the problems raised by the limited amount of information on small screens [May and Baddeley 2006; Sanneblad and Holmquist 2006].

Zoomable user interfaces (ZUIs) and pen/finger gestures have been designed to improve the usability of small devices applications [Bederson et al. 2004; Karlson et al. 2005]. Zooming technology is becoming popular in new portable systems, like Microsoft Windows Mobile 6.0 devices and the Apple iPhone. Native SDK and third part API, like the *Piccolo* framework (http://www.cs.umd.edu/hcil/piccolo/), provide standard widgets and gesture recognition modules and, in some cases, allow software developers to easily extend the interaction framework of the device and of the operating system with custom components.

The applications built with such interfaces, while sharing with our model the smooth transition between overview and detail view on user data and services, do not help the user to change the perspective in a consistent way while browsing data. According to our vision, the system should allow the user to select information and services belonging to a given category and to present, on a synthetic view, the related items belonging to other categories. Each item should be accessible in a detail view and used as the focus of a new perspective in a highly dynamic fashion.

In the area of geolocalized information management, exploited in mobile systems, there is a wide set of components and frameworks to bind location information to services, ranging from the simple task of requesting a map to a server through the web via http request or through web-services intermediation, to rendering geographic locations with different content styles, to the integration with retrieval services. A significant number of these frameworks aims to enable the building of applications for the major smartphone operating systems, bridging the gap between the differences among the platforms [W3C 2009; Google Inc. 2009; OMTP Ltd. 2009; Rhomobil Inc. 2009].

3. SITUATED USER ACTIVITIES

The management of personal activities during day by day professional life is a good example of a mixture of the Activity Theory and the Situation Cognition approaches. In planning such activities a person flows through a set of information that is in part objective (facts, regulations and practices), in part subjective (judgements, feelings, history, experience) and builds an initial version of a plan. The more the activity is clerical, the longer the plan will survive as new information is acquired. However, decision based activities typical of managers and professionals are less stable, and need a higher level of adaptation to changing situations. In such cases an initial plan is roughly sketched and is adapted and completed as the activity evolves. Information search is an essential part of such activities; information access is not regulated by a series of independent, atomic needs, but by a complex network of interrelations that link the activity execution to the situation changes.

Using popular terms, information is rarely *queried*, indeed it is mostly *browsed* according to the user context, which is often known (or sensed) by the user but can hardly be formalized. Browsing, however, is not completely free, since it is driven by the evolving situation, therefore based on previously acquired information. The result is a controlled navigation among services and information, where the user decides on the fly which are the relevant paths to follow for fulfilling the activity tasks.

Such a behavior assigns to the user the role of *service composer*, designing a scenario very different from (almost opposite to) the automatic service discovery and composition of pervasive computing. The system may help the user evidencing the connections between people, locations and services belonging to separate activities. The knowledge of elements overlapping several activities (e.g., a specific location shared by different tasks) may become relevant if the situation changes; such elements, if properly evidenced, may act as switching points that can suggest the user to work on another activity or on another task .

The active role of the user in service selection in our vision is not a negative feature but a necessity, due to the variability of the cases the user needs to face. Three important issues stem. First, our scenario is sampled on a user performing activities related to personal information management and day by day organization of own professional and social life. Both are subject to a reasonable overview planning but must adapt to events and constraints set by other people with whom the user relates. Second, in a PDA-based information management services are general purpose and largely independent, even if collected in suites. This is due to technical limits of program integration in small capacity devices and to the need of supporting a wide spread of use habits. Third, a considerable part of information needs is resolved by searching the Web, and the retrieved information is used in the services useful for performing activity related tasks.

As to this last issue, Web 2.0 has made popular and widely available a fan of services for sharing social information, for managing maps and locations, for accessing categorized information, for publishing personal and professional profiles, and so on. It is common habit to access such services through a generic query to a search engine such as Google or Yahoo, whose answer is used as a link to the service.

In the personal information management framework the services available to a user are split in two classes: generic and specialized services (i.e., programs). Generic services like agenda, address book and messaging, in principle are not related to the fulfillment of a specific task but are used as commodities, much as generic libraries in a programming environment. Specialized services are in principle related to the fulfillment of a task, such as map-based information discovery. The distinction is fuzzy, and double-faced services are common: for example, mail can be considered a generic services when reading new mail (since the content is unexpected), but is a specialized service targeted to a specific use when replying to a message, or when writing a new message related to a task. In both cases, some mail parameters are bound to the task, e.g., the receiver name and the subject (which are automatically filled in case of reply) or a template of the message, prepared by a task manager program.

A system supporting personal activities should:

(1) Know about user tasks and activities configured by the user, freely set by the user, possibly according to templates that represent initial, modifiable, activity plans.
(2) Track the status of tasks and activities, and track the set of information associated to the tasks.
(3) Present to the user, besides the standard interface to operations and services, a set of information and services related to the user situation as described by the

activities and tasks status: services strictly functional to a task could be initiated by the system by suggesting the user an order of execution, and by filling variable data with information taken from the situation. In other words, the system should explicitly support the tasks that are related to the specific situation.

(4) Display the relations between the activities and the tasks defined by the user, that might have been defined at different times. Merging data related to different situations, with the explicit synthetic representation of the contact points, is useful to give a higher level of knowledge, mostly effective if the situation changes. The linking information (i.e., people, locations and services shared by different activities) may act as switching nodes if the situation changes, presenting immediately a set of starting points for beginning another activity.

We propose to support this type of operations with a model, called *Situation Lens*, composed of a schema (a simplified ontology) for describing activities, tasks, services and situations, of an architecture mapping the schema to the PDA personal information management system, and an operational interface.

Through the interface the user is able to explore the information and the services contributing to his/her situation, to receive an overview of the situation or to focus on the different information objects, to have details on demand, to cover different areas of the personal data, and is supported in the composition of personal services according to the tasks to be executed.

4. THE SITUATION LENS MODEL ONTOLOGY

The model of activities and services on which the Situation Lens is grounded is close to the *situation metaphor* defined by Hewagamage and Hirakawa [2000]. It is built around four classes of objects: activities, tasks, events and services.

Activity. An activity is a set of tasks, $act = \{t_1, t_2, \ldots\}$, partially ordered. The set may change during the activity execution: new tasks can be added, some tasks can be removed, thus making the activity to evolve from an initial state act_0 to a final state act_f according to the user needs. The initial state represents the user needs as defined at activity planning; the set of tasks is a plan for the activity and is also its history when tasks are done. The initial state of an activity can be empty, $act_0 = \{\}$, semantically representing a placeholder for an activity anticipated by the user but not planned at all.

Task. A task is a tuple $t = (task\_name, description, ES, status)$ where

- *task_name* is a unique task identifier;
- *description* is a description of the task (we do not enter into details);
- *ES* is a set of pairs (e_i, s_i), meaning that upon occurrence of event e_i service s_i can be executed.
- *status* is an enumerative value (*to_do, in_progress, done*), which can be set by service execution. Events occurring after task completion have no effect.

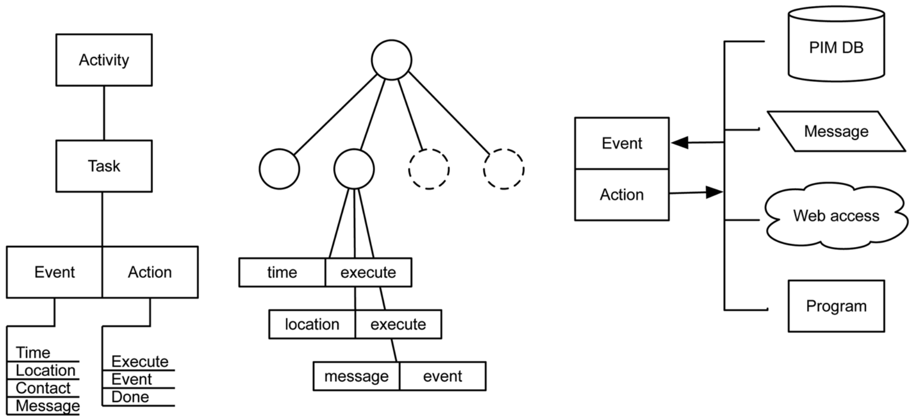

Figure 1. The Situation Lens Model.

In general a task can be fulfilled by different sets of services, according to the user judgment, therefore the pairs of E need not to be disjoint, leaving the user the responsibility to execute a service upon an event.

Event. An event is a tuple $e = (event\_name,\ t,\ l,\ c,\ m)$ where

- t is a time specification (value or interval);
- l is a location;
- c is a contact;
- m is a message.

Time, location, person and message details are not relevant here. An event can occur at a specified time, during a time interval, upon a message arrival, or can be associated to a contact (e.g., adding a new contact triggers a new event). Any of t, l, c and m can be empty, but at least one must be specified.

Service. A service is a tuple $s = (service\_name,\ description,\ A)$, where A is a set of actions. Each action is one of:

- *execute*: execution of a service through a program;
- *create_event*: creation of an event specification;
- *change_status*: change of the task status.

Services are executed by programs through which the user accesses personal data, creates and sends messages, adds contacts, writes notes (which are the typical applications of a PDA program suite), and accesses external services in the shape of an URL fed to a browser.

We do not enter into details since our aim is not to provide an application model but only to model the relationships between information, applications and user activities. The concepts described above are independent from the specific implementation and may be applied to different platforms, mapping and refining events and services on the actual PDA software environment. In the definition of an event, time and location

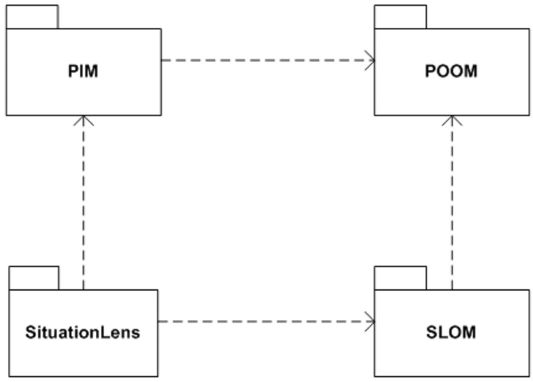

Figure 2. The Situation Lens Architecture.

define at the minimum extent the *objective* user situation, while contacts and messages are the minimal interface with external information, defining the *subjective* user situation. Other components may extend the model, but do not add relevant concepts for the model goals. The same holds for service actions, which are described as a core that can be extended at need.

The model defines a set of classes from which data instances can be derived, linked to services and information managed by the user applications. In Figure 1, the activity and service model is on the left. The centre diagram depicts the instances: tasks to be completed are represented with dashed borders; task instances are event-action pairs, where actions can generate new events, which, in turn, drive the user to add new tasks. On the right, the database of the Personal Information Manager (PIM) collects data about user contacts, agenda, notes, maps, etc.; messages, web access and external programs represent the prototypes of services the user can activate to fulfill his/her activity; events are triggered by information contained in the PIM or by the user intervention, who interprets the messages, the browsed information and the result of the executed programs.

Situation. The data instances describe a state of the user activity. Instances taken at different times describe the evolution of the user activities. A snapshot of the data instances is a *situation*, the description of the information the user has about the current status of his/her activity. Snapshots taken at different times describe the user history, as in the model proposed by Hewagamage and Hirakawa [2000].

5. THE SITUATION LENS SOFTWARE ARCHITECTURE

The Situation Lens model has been applied to extend the behavior of the PIM of the Windows Mobile systems. The implementation has been done in the version 5 of Windows Mobile.

As stated in the introduction, the Windows Mobile platform [Yang et al. 2006] has been chosen because of its diffusion and the possibility to enhance its personal information model with new informative objects and relations, still using the standard informative objects for additional operation such as feeding the database, modifying

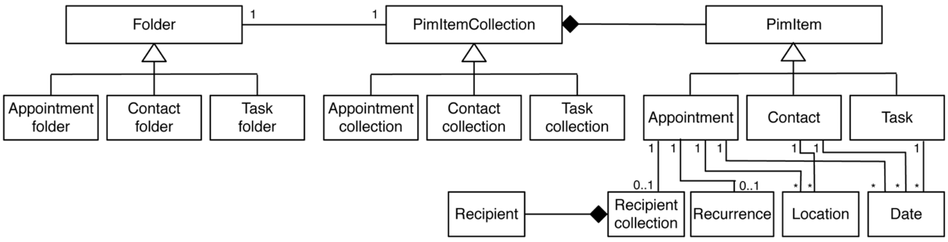

Figure 3. The Pocket Outlook Object Model.

it or exporting data for other uses. Such possibility is not easily granted by other platforms, such as the Apple iPhone [Apple Inc. 2009], as we shall discuss in Section 7. The integration with the desktop platform is an additional feature of the Windows Mobile platform that improves the system generality and may also account for the definition of additional interfaces, more suited to large screens and to different interaction devices.

The architecture is illustrated in Figure 2; it includes:

- the *Pocket Outlook Object Model* (POOM) which is the Windows Mobile data model for the management of the user's personal information [Microsoft Corporation 2009];
- the set of standard software functionalities, API and widgets, that allow PIM users to query and to visualize POOM instances;
- the *Situation Lens Object Model* (SLOM), the actual implementation of the Situation Lens model ontology described in Section 4, which acts as a wrapper of the underlying POOM;
- the *Situation Lens* application, which through the SLOM and the embedded functionalities of the PIM gives the user a representation of the situation and the ability to manage the entities rooted on the POOM and extended by the SLOM.

5.1 The Pocket Outlook Object Model

Figure 3 shows a fragment of the *Pocket Outlook Object Model* (POOM). The model is strongly based on the concept of *collection* for representing and storing the data instances. As such, while operations to manage items are easy and uniform (e.g., adding, removing and sorting items), the relations among data belonging to different collections are not apparent from this schema. Indeed, only the right part of the diagram introduces classes and attributes for the data relevant for the user: the *Appointment*, *Contact* and *Task* entities and their sub-entities. Manipulation function are provided for such data that allows the user to create, edit and delete instances of appointments, contacts and tasks.

In the Situation Lens model implementation we have chosen to reuse, when possible, the native set of functions provided by the POOM, to preserve the user knowledge about the Windows Mobile widget operations.

5.2 The Situation Lens Object Model

The Situation Lens Object Model (SLOM) is the actual implementation of the

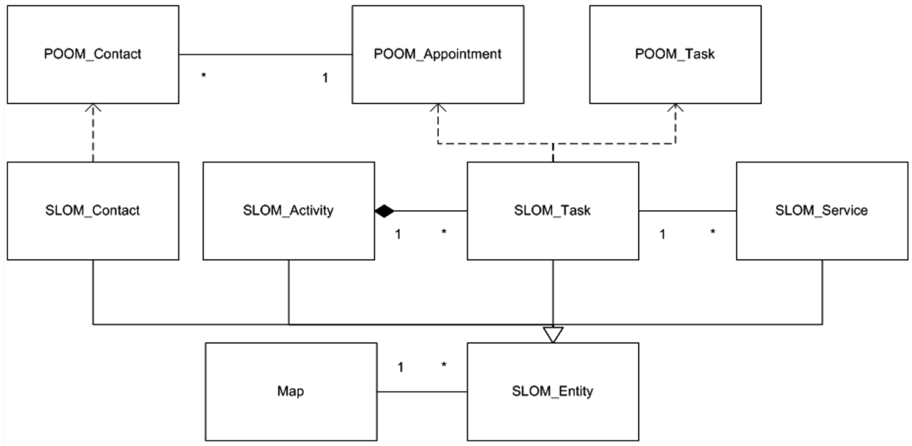

Figure 4. The Situation Lens Object Model.

Situation Model ontology described in Section 4, illustrated in Figure 4 by an Entity-Relationship diagram. SLOM adds new entities to the POOM and extends pre-existing entities with new attributes. It aims to enhance the user navigation in POOM entities, relating them according to the activity, task and situation defined in Section 4.

The definition of new entities and relations, possible due to the accessibility of the Windows Mobile POOM primitives, is of fundamental importance for the interface proposal, that relies on the visualization and navigation of a rich set of informative objects. The associated set of relations allows the user to seamlessly change the focus on the personal information domain, selecting any informative item of interest for getting an immediate visual feedback about the other items related to it in the proper perspective. The process, interactive and iterative, allows the user to satisfy different information needs, ranging from known item search to exploratory seeking.

SLOM introduces the following entities not present in POOM:

- *Activity*: a set of tasks, as described in Section 4.
- *Service*: the atomic step of a task execution, corresponding to an action of the user triggered by en event.
- *Map*: data defining the location of a generic entity in a geographical coordinate system; we refer to the Google Maps environment due to the possibility to switch between static and dynamic maps according to the connection status of the device, as described in Section below; the model is however largely independent from the specific geographic information supplier.

SLOM also extends the POOM *Contact* entity definition and defines a new *Task* entity that includes both the role of a POOM task and of a POOM appointment.

The situation is an instance of SLOM where each entity instance refers directly or indirectly to an instance of a POOM entity. In this way SLOM is able to manage wider information than POOM while preserving the content and consistency of POOM instances.

As in many models of user context and situation, location plays a distinguished role, since it is related to the human need to execute tasks by accessing physical resources and moving in the space. Mobile applications exploit such a need by giving the user the ability to identify the relations between the space and the tasks in an automatic way. Hence, in SLOM positional information is associated to generic entities, in particular to the following entities:

- *Service*: the location L of a service S defines a *Service Access Point*, which can corresponds to a context aware localized service, such as a data synchronization service between the PDA device and the office personal computer, or a service available in a wide area, such as an information service available in a railway station.
- *Contact*: one or more locations can be defined to bind geographical information to a contact (i.e., home, office, etc.).
- *Task*: the location of a task T is related to the location of the Service Access Point and refers to the logistic aspects the user should consider in order to accomplish the task.

5.3 The Situation Lens Operational Interface

The Situation Lens acts as an aggregator and a dynamic filter on the data managed by the Windows Mobile PIM and on the available services; it generates a situation snapshot according to three different perspectives:

- a *global perspective*, showing a synthesis of the user activity: a qualitative view of the activity status and of the entities involved is displayed as an index to explore the activity components; the global perspective interface is the key for changing effortlessly the focus of the user interest;
- a *task perspective*: the part of the user situation related to a specific task is given, acting both as a summary of the task and as an index to the task details;
- a *detail perspective*: the user can explore detail data extracted from the personal information database and execute programs and services on such data.

From the application development side the Situation Lens is supported by the software components developed for managing the POOM data.

Figure 5. The Task and Detail Perspective Interface.

5.4 Mapping the Situation to the Services

In the Situation Lens framework, the situation is mapped to the POOM through an extension of the entity *Task*, which in the POOM is simply a description with a status and possibly a deadline. Indeed, the POOM task is similar to an action in the Situation Lens model. In our model tasks are containers of references to other PIM items, collecting index information about contacts, appointments, actions, messages, and maps. Figure 5 shows an example of a *task* perspective and two *detail* perspectives. The screenshot on the left shows the task description, which is an index to detail information. Tapping an item opens a detail perspective view in the PDA service that creates and manages that type of data. The screenshot in the center of Figure 5 shows the information of one of the two contacts of the task, while the right screenshot shows a static map centered on the location associated to the task. The PDA services show their own interface and behavior, as if opened directly from the *Program* menu of the PDA.

Maps are defined at SLOM layer while services are separated; they build

- the reference to a device software component (i.e., an application for writing text);
- the reference to an Unified Resource Locator (URL) (i.e., a web portal of a specific flight booking company);
- the reference to a service access point (i.e., a railway station wi-fi access point).

Switching between services preserves the situation since service activation is always done through the task description, which filters the instances of data accessed by specific services.

5.5 The Situation Lens Overview Interface

Figure 6 illustrates the interface of the situation lens *global perspective*, offering to the user an overview of the situation related to an activity. The interface has been designed with two goals in mind: (1) to provide an immediate perception of the task complexity through a summary view of the entities involved, and (2) to allow the user to explore the relations between the entities and the relations with other tasks and activities, masking the details.

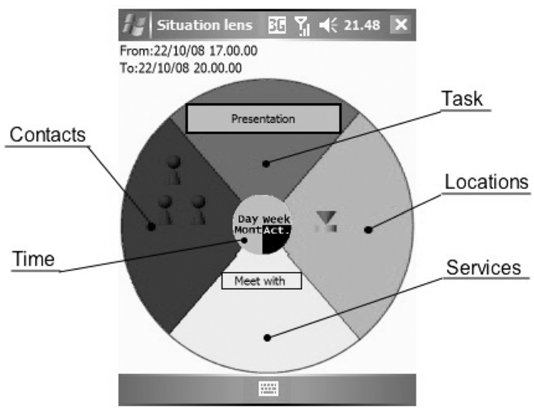

Figure 6. The Situation Lens Global Perspective Interface.

The user situation is represented as a circle split in four parts, related to the tasks, the locations, the services and the contacts. The user "rotates" the lens so that the current perspective, i.e., the focus of interest, is in the top position. A selection of an instance of a task, location, service or contact reveals in the other quadrants an iconic view of the information items linked to the instance. A central section defines the temporal zoom of the lens, from the current day to the whole time span of the activity; time selection coherently updates the four quadrants' content.

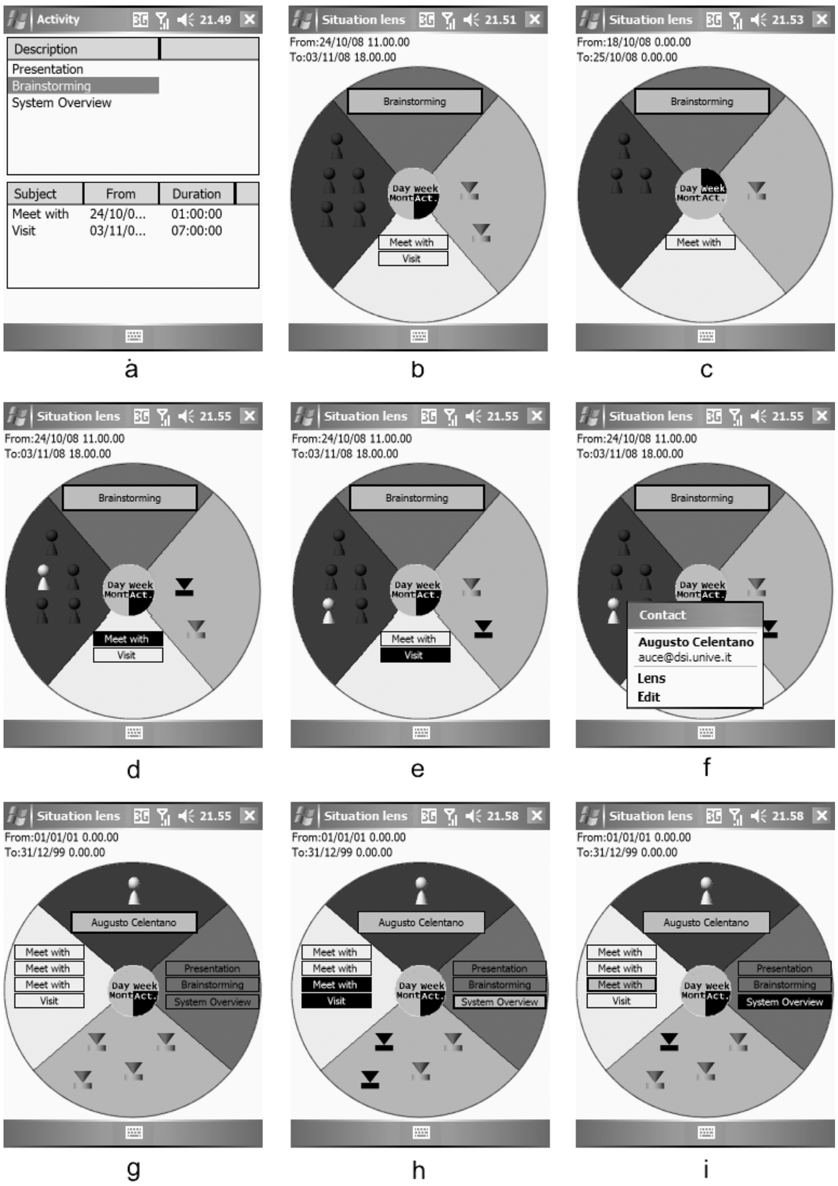

Figure 7. A Sequence of Operations on the Situation Lens Global View.

In Figure 7, a sample sequence of operations on the global view is shown. The user selects a task from the activity screen, as shown in Figure 7a. The global view opens with the task as the focus information (Figure 7b), the time control set initially to the whole activity. The number of icons in the other quadrants matches the number of instances of locations, services (i.e., actions) and contacts for that task and for the set time span.

Setting a different time span, e.g., a week, as in Figure 7c, projects the situation by showing only the items relevant for the current week.

Tapping on an item (an icon) identifying a person, a location or a service selects the item (in grey) and shows the association between that item and the other items of the situation: in Figure 7d and Figure 7e the tasks, locations and services related to two different contacts are shown in black.

A tap and hold gesture on an icon shows identification labels and a menu for information filtering and direct access to the detail view, as shown in Figure 7f.

The selection of the *Lens* operation in the menu changes the Situation Lens focus to the object currently selected.

While the gesture of tapping on an item allows the user to view which item, among ones the already visualized, has a specific relation with the one selected, leaving the overall layout untouched, the selection of the *Lens* operation on an item causes a reorganization of the global perspective interface. The selected object becomes the only representative of its class and is placed on the upper part of the circle, to emphasize that it is the current center of interest. The other circle sections are filled with a (possibly new) set of objects associated to the current center of interest. Such operation can be done iteratively to explore the network of relations associated to the informative objects contained in the personal information database.

Figure 7g shows a different perspective where the contact is the focus information, and the situation is shown under the perspective of the contact, showing the tasks in which it is involved, the locations and the services associated to the tasks. Selecting a task, as in Figure 7h, shows the associated locations and services, and selecting a service, as in Figure 7i, shows the tasks in which the service is needed.

6. A PROTOTYPE IMPLEMENTATION

We have developed a prototype implementation of Situation Lens to validate the design of the SLOM-POOM interface and the adequacy of the perspectives offered by the operational interface, in particular of the global perspective view. In the prototype we have implemented most of the features described in the previous sections. In particular, the graphical user interface (1) visually represents the user situation, (2) offers the widgets the user needs to handle SLOM data, and (3) allows the user to access all the software components related to a service. As already noted, the prototype has been implemented in the Windows Mobile 5 System. Two devices have been tested with different screen resolutions: a Dell Axim X51v PDA and an HTC P3600 smartphone. The screenshots in Figures – are taken from the HTC prototype.

6.1 Widgets to handle the SLOM data

The Situation Lens GUI developed for SLOM data handling consists of a set of forms for data access and editing. The appearance of each form has the same look and feel of the standard user interface of the Windows Mobile functions for PIM data management.

Activity management. As described in Section 4, an activity is defined by a set of tasks; at the interface level the user defines a name for an activity and creates an initial set of tasks. The time span of the activity is synthesized from the earliest start time and the latest end time of the set of tasks.

Task management. A SLOM task has a number of properties: (1) it has a time execution or deadline specification; (2) it can be related to one or more contacts; (3) it can occur on a specific location; (4) it could require a set of services. Exceptions are allowed, e.g., tasks without a time specification act as generic reminders; they do not contribute to the definition of the activity duration, and leave the user completely free to decide about their completion.

A SLOM task instance extends the task concept in the underlying POOM. Items (1) and (2) above cover the definition of a POOM appointment, while items (3) and (4) link a task with a location that can be managed by geographical services, and with the execution of services provided by the mobile device or accessed via network connection. Maps and service management widgets are described later in this section. The service needs of a task are defined by the user as an heterogeneous collection of references to instances of SLOM services.

Map management. Maps are implemented via the Google Maps Static APIs [Google Inc. 2008], a simple but effective way to display ancillary information as geographic maps that allows the application to receive maps on the client side with simple calls to HTTP request methods.

If a network connection is active on the device, the Situation Lens application is able to get the map of a specific geographic area and to store it in the non volatile memory of the device. If no network connection is available, only maps already stored can be viewed as geographical locations of SLOM entities.

Service management. As described in Section 5, a service is a local application, an URL or a geo-located service access point, defined by the user when drawing the initial plan for an activity. Services can be added subsequently as reactions to events related to the situation; in particular, services can be attached to a task through a contact details, such as an e-mail address or a phone number, allowing the user to execute trivial actions like sending a message or posting a call with the same mechanism used for controlling more complex services.

7. DISCUSSION

Many interfaces related to the information visualization can be described by the well-known Shneiderman's *mantra*: *Overview First, Zoom and Filter, Details on Demand* [Shneiderman 1996]. As evidenced by Craft and Cairns [2005], the Shneiderman mantra is rather descriptive than prescriptive and, as stated by Shneiderman himself,

Table I. A comparison between Shneiderman's Model of Interaction and the Situation Lens Interface Functionalities.

Overview first	
SH	representation of all the objects in a single screen; such visualization is the basis for exploiting the second functionality (Zoom and Filter)
SL	representation of all the objects in a single list; the user must choose a specific object for proceeding
Zoom	
SH	zoom in and zoom out
SL	not available in the current version of the interface
Filter	
SH	usually different sliders are available
SL	a time slider is available
Focus	
SH	not available
SL	objects may reveal relations with other classes of objects; visualization of relations constrained to the objects of the current selection is also possible
Details on demand	
SH	representation of the details of a specific object
SL	same functionality

is targeted to wide high-resolutions screens. While the solution proposed in this work can partly be described in Shneiderman's words, there are some differences due to the different types of information and to the different types of device chosen for the information delivery, which are small screen portable devices.

Table I compares the functionalities of the visualization interfaces compliant with the model suggested by Shneiderman's mantra and the Situation Lens model.

The main functionality of the Shneidermsn's model is the *overview*, the possibility to visualize, in a single high-resolution screen, all the items belonging to a given set. The items are characterized by a number of properties, shared by all the items, that can be used in a second phase to *filter* information, showing to the user only the objects compliant with the filter. The user may also zoom, in order to focus his/her attention on a part of the objects (*zoom in*) or to be aware of the relations with the neighbors (*zoom out*). Finally, the user may ask details of a specific object of interest (*details on demand*).

The interface of the Situation Lens prototype allows the user to reach the same level of information details, but the path to get such result is different. The small size of the screen of mobile devices constrains *ex ante* the amount of data that can be displayed in a global, overview panel. Therefore, we decided to limit it to a list of the main informative objects of the task, each identified with a label. The user must choose a specific task to start with, to access the other views, available on the layout illustrated in Figure 7.

Functionalities available in this layout can be mapped to the Shneiderman model only partially. In particular, in the current version of the interface, information available on this layout may not be zoomed in and out (but it may be implemented in future versions, coupled with specific gestures for allowing the user to see more details, such as the name of a location or of someone involved in the examined task). Filtering is available, but, as shown in Figure 7, is limited to the use of a time selector for enlarging or restricting the focus on specific temporal slices.

Additional functionalities of the Situation Lens are missing in the Shneiderman's visualization model. Such functionalities are applied to a different informative scenario. In the Shneiderman's model, only the main objects are visualized in the main visualization areas, and their properties are represented as widgets positioned outside the main visualization area for showing or hiding the main objects. In our approach, most properties of the main objects are visible as additional objects positioned in the main visualization area. Such representations are not only objects revealing additional details on demand, but they can also be used as handles to put information in a different perspective.

Such *subjective* perspective view represents a visualization mode where the lens focus is fixed on a primary object and all the related objects are shown at a distance. Such a relational visualization mode is opposed to the Shneiderman's model that, instead, relates to a zoomable bird-eye view.

In addition, the focus on a specific handle may also be used to reveal relations among the current selection of objects.

Concluding, while our approach shares with the Shneiderman's model the dynamic representation and part of the functionalities, it presents additional advantages that may be summarized in the ability to represent the situation focusing on specific objects, belonging to a more heterogeneous visualization scenario, to see things according to different, subjective, perspectives.

Compared to the situated metaphor of Hewagamage and Hirakawa [2000], our approach is less complete but more flexible and more oriented to situation facets linked to the user perception and goal rather than physical status. Hence, the diminished possibilities of planning and automatically executing services concern only the observable parameters of the user status. Our approach gives more importance to the decisions the user takes according to a personal and subjective evaluation of the objective information at hand.

A comparison of the Windows Mobile platform with the Apple iPhone platform, which is based on the Mac OSX operating system, reveals a quite different philosophy in the two systems which is not restricted to the user interface, even if it is the most visible component. The iPhone operating system[1] does not allow programs to share information directly; each application can access all local information in a *sandbox* which is protected by an opaque naming mechanism to prevent other applications to access it. Data sharing can only be accomplished through a set of APIs of the iPhone Software Development Kit [Apple Inc. 2009] providing in a controlled way an interface to data of the standard applications. In the current version of the operating system, APIs are provided for handling data of the Contacts application but no access is provided to the Calendar application data, i.e., tasks, meetings, etc.. Then, any PIM application and specifically a potential Situation Lens prototype on this platform

[1]This discussion applies also to the iPod touch device that shares with the iPhone most of the features, except telephony and geographical positioning

could not provide global, integrated view of the user situation unless the whole set of data is replicated and managed directly by the new application.

Generally speaking, mobile technologies seem to follow two different trends:

- one which offers to the developers community a full APIs access to the personal information set, which has the advantage of developing applications fully integrated with the embedded device software suite, providing the ability to synchronize data with desktop computer and other mobile devices with native solutions;
- one which offers a restricted API access to data managed by native embedded applications, limiting the potential development of new integrated applications.

Both trends reveal pros and cons: the former approach opens data to the development of efficient and user friendly applications, but might raise privacy and security problems, as explicitly noted in the iPhone SDK documentation introducing the application *sandbox* concept, while the second approach, while preventing security leaks, depends on the richness of the interface provided by the native applications, which are developed each with specific goals in mind that could not support designers to integrate them.

Recently, new technologies such as Google Calendar are offering a web based architecture of personal information management, suited with a set of mobile client application and web application which try to offer a powerful set of PIM functionalities and sometimes an integration (through synchronization) with embedded platform databases.

8. CONCLUSION

Two issues are relevant in designing future extension of the Situation Lens model and of its interface: the relations between personal and social data, and the increasing availability of new devices supporting multitouch gestures; even if projected on different axes of interest, they are related by the growth of applications and devices for enhanced, easy participation of people to a shared social life.

While our current work has been focused on personal activities, the data structure, the architecture and the interface presented are also adequate for supporting social interaction. In particular, the properties associated to the task, the main informative object in the perspective of professional activities, may help social communication, since it manages personal details for contacting people involved in shared activities, and information about the people location. The awareness of the presence of a colleague or friend in a location nearby may be an important factor that can be supported by our architecture, triggering the choice of the next actions to perform, thus leading to an overall reorganization of the daily activities. Of course, while the availability of a database shared between people involved in tasks to perform opens additional possibilities for social interaction, it raises additional issues, mostly related to consistency and to privacy, that are outside the scope of this work and that need to be considered in future developments.

Also, the new generation of multitouch devices like the Apple iPod touch and iPhone open new possibilities for natural gesturing, assuming the restriction about data interoperability noted in Section 7 will be removed. To *rotate* a lens or to *enlarge* a view are no longer metaphorical actions but real physical gestures. Our early experiments show that switching between entities, even with conventional gestures, improves the user perception of the relations among the tasks, mainly for activities long lasting or frequent, often bound to a common base of social and professional relations. Multitouch gestures could improve the perception of continuity between different views on complex personal data, as during the manual exploitation of everyday tasks.

REFERENCES

APPLE INC. 2009. Iphone reference library. http://developer.apple.com/iphone/library/navigation/index.html.

BARDRAM, J. 1997. Plans as situated action: An activity theory approach to workflow systems. In *ECSCW, Fifth European Conference on Computer Supported Cooperative Work*.

BEDERSON, B. B., A. CLAMAGE, M. P. CZERWINSKI, AND G. G. ROBERTSON. 2004. DateLens: A fisheye calendar interface for PDAs. *ACM Trans. Comput.-Hum. Interact. 11,* 1, 90–119.

BENNETT, F., T. RICHARDSON, AND A. HARTER. 1994. Teleporting—making applications mobile. In *Proc. IEEE Workshop on Mobile Computing Systems and Applications*.

BERHE, G., L. BRUNIE, AND J. M. PIERSON. 2005. Distributed content adaptation for pervasive systems. In *ITCC 2005, International Symposium on Information Technology: Coding and Computing*. Las Vegas, Nevada, USA, 234–241.

BERTELSEN, O. AND S. BODKER. 2003. Activity theory. In *HCI Models, Theories, and Frameworks: Toward a Multidisciplinary Science*, J. Carroll, Ed. Morgan Kaufmann, San Francisco, 291–324.

BOLCHINI, C. AND E. QUINTARELLI. 2006. Filtering mobile data by means of context: A methodology. In Proc. OTM Workshops 2006 - Context-Aware Mobile Systems. *Springer-Verlag, LNCS 4278*, 1986–1995.

BOLCHINI, C., F. A. SCHREIBER, AND L. TANCA. 2007. A methodology for very small database design. *Information Systems 32,* 1 (March), 61–82.

BOLL, S., W. KLAS, AND J. WANDEL. 1999. A Cross-Media Adaptation Strategy for Multimedia Presentations. In *ACM Multimedia*. 37–46.

BRADLEY, N. A. AND M. D. DUNLOP. 2005. Toward a multidisciplinary model of context to support context-aware computing. *Human-Computer Interaction 20,* 4, 403–446.

CELENTANO, A. 2008. Layered context modeling in situated information processing. In *DMS 2008, Int. Conf. on Distributed Multimedia Systems*. Boston, MA, USA.

CELENTANO, A. AND F. PITTARELLO. 2007. Situated multimodal documents. In *MIMIC '07, 1st Int. Workshop on Management and Interaction with Multimodal Information Content*. Regensburg, Germany.

CHAARI, T., F. LAFOREST, AND A. CELENTANO. 2007. Adaptation in context-aware pervasive information systems: The SECAS project. *International Journal on Pervasive Computing and Communications 3,* 4.

CHEN, G. AND D. KOTZ. 2000. A survey of context-aware mobile computing. Tech. Rep. TR2000-381, Dartmouth College, Department of Computer Science.

CLANCEY, W. J. 1997. *Situated cognition: on human knowledge and computer representations*. Cambridge University Press.

COSTA, P. D., L. F. PIRES, AND M. VAN SINDEREN. 2005. Architectural patterns for context-aware services platforms. In *Proc. of the 2nd International Workshop on Ubiquitous Computing, IWUC 2005*. Miami, FL, 3–18.

CRAFT, B. AND P. CAIRNS. 2005. Beyond guidelines: What can we learn from the visual information seeking mantra? In *IV '05: In Proceedings of the Ninth International Conference on*

Information Visualisation. IEEE Computer Society, Washington, DC, USA, 110–118.
DAVIES, N., K. CHEVERST, K. MITCHELL, AND A. EFRAT. 2001. Using and determining location in a context-sensitive tour guide. *IEEE Computer 34,* 8, 35–41.
DEY, A. 2001. Understanding and Using Context. *Personal Ubiquitous Computing 5,* 1, 4–7.
DOURISH, P. 2000. A foundational framework for situated computing. Position paper for the CHI 2000 Workshop on Situated Computing: A Research Agenda.
ELOUAZIZI, N. AND Y. BACHVAROVA. 2004. On cognitive relevance in automatic multimodal systems. In *Proc. IEEE Conf. on Multimedia Software Engineering*. Miami, FL, 418–426.
GERO, J. S. 2003. Situated computing: A new paradigm for design computing. In *CAADRIA03*, A. Choutgrajank, E. Charoenslip, K. Keatruangkamala, and W. Nakapan, Eds. Bangkok, 579–587.
GOOGLE INC. 2008. Google static maps api. http://code.google.com/apis/maps/.
GOOGLE INC. 2009. Google code, javascript geo location framework for the mobile web. http://code.google.com/p/geo-location-javascript/.
HEWAGAMAGE, K. P. AND M. HIRAKAWA. 2000. Situated computing: A paradigm to enhance the mobile user's interaction. In *Handbook of Software Engineering and Knowledge Engineering*, S. K. Chang, Ed. World Scientific.
JAMESON, A. 2001. Modeling both the context and the user. *Personal Technologies 5,* 1, 29–33.
KARLSON, A., B. BEDERSON, AND J. SANGIOVANNI. 2005. AppLens and LaunchTile: Two designs for one-handed thumb use on small devices. In *Proc. of SIGCHI Conference on Human Factors in Computing Systems*. 201–210.
LEMLOUMA, T. AND N. LAYAÏDA. 2004. Context-aware adaptation for mobile devices. In *Proc. IEEE Int. Conf. on Mobile Data Management*. Berkeley, CA, 106–111.
LUEG, C. AND R. PFEIFER. 1997. Cognition, situatedness, and situated design. In *Proc. of Second International Conference on Cognitive Technology*. Aizu-Wakamatsu City, Japan, 124–135.
MAY, R. AND B. BADDELEY. 2006. Visual analytics: Large and small display environments. In *Information Visualization and Interaction Techniques for Collaboration across Multiple Displays, Workshop associated with CHI'06*.
MICROSOFT CORPORATION. 2009. Pocket outlook object model (poom). http://msdn.microsoft.com/en-us/library/aa914277.aspx.
NARDI, B. A. 1996. *Context and Consciousness: Activity Theory and Human-computer Interaction*. The MIT Press, Cambridge, MA.
OMTP LTD. 2009. Bondi. http://bondi.omtp.org/.
PETRELLI, D., E. NOT, M. ZANCANARO, C. STRAPPARAVA, AND O. STOCK. 2001. Modelling and adapting to context. *Personal Ubiquitous Comput. 5,* 1, 20–24.
RHOMOBIL INC. 2009. Rhodes. http://rhomobile.com/products/rhodes/.
SANNEBLAD, J. AND L. E. HOLMQUIST. 2006. Ubiquitous graphics: combining hand-held and wall-size displays to interact with large images. In *AVI '06: In Proceedings of the working conference on Advanced visual interfaces*. ACM, New York, NY, USA, 373–377.
SCHMIDT, A. 2006. A layered model for user context management with controlled aging and imperfection handling. In *MRC 2005, Modeling and Retrieval of Context*. LNCS, vol. 3946. Springer Verlag, 86–100.
SCHMIDT, A., M. BEIGL, AND H.-W. GELLERSEN. 1999. There is more to context than location. *Computers and Graphics 23*.
SEEL, N. M. 2001. Epistemology, situated cognition, and mental models: 'like a bridge over troubled water'. *Instructional Science 29*, 403–427.
SHNEIDERMAN, B. 1996. The eyes have it: A task by data type taxonomy for information visualizations. In *VL '96: In Proceedings of the 1996 IEEE Symposium on Visual Languages*. IEEE Computer Society, Washington, DC, USA, 336.
STRANG, T. AND C. LINNHOFF-POPIEN. 2003. Service interoperability on context level in ubiquitous computing environments. In *Intl. Conf. on Advances in Infrastructure for Electronic Business,*

Education, Science, Medicine, and Mobile Technologies on the Internet. L'Aquila, Italy.
STRANG, T. AND C. LINNHOFF-POPIEN. 2004. A context modeling survey. In *UbiComp, 1st International Workshop on Advanced Context Modelling, Reasoning and Management*. 34–41.
SUCHMAN, L. 1987. *Plans and Situated Actions*. Cambridge University Press, New York.
TANCA, L. 2007. Context-based data tailoring for mobile users. In *Proc. BTW Workshops 2007*. 282–295.
W3C. 2009. Geolocation api specification - working draft. http://www.w3.org/TR/geolocation-API/.
YANG, B., P. ZHENG, AND L. M. NI. 2006. *Professional Microsoft Smartphone Programming*. John Wiley and Sons.

Augusto Celentano is full professor of Information Systems at Università Ca' Foscari Venezia, Italy. He received a Master Degree in Electronic Engineering from Politecnico di Milano. He has been Deputy Rector for information technology at Università Ca' Foscari Venezia, member of scientific committees in research and educational centers of the Politecnico di Milano, and consultant and scientific coordinator of research projects of the European Union. His current research interests are in mobile, pervasive and context-aware information systems, in multimedia systems, in Human Computer Interaction, and in cultural heritage applications.

Stefano Faralli has been assistant researcher at the Computer Science Departments of Università di Roma Sapienza and Università Ca' Foscari Venezia, Italy. He received a Master degree in Computer Science from Università di Roma Sapienza. His research interests are interactive systems for live performances in arts, body postures detection and multimedia production.

Fabio Pittarello is a researcher at the Computer Science Department of Università Ca' Foscari Venezia, Italy, where he teaches web design and human computer interaction for the computer science curriculum. His current research interests include both theoretical and application oriented aspects of HCI in web-based and ubiquitous contexts, with a particular reference to interaction in 3D environments, adaptivity, multimodality and universal access to information.