The Situation Lens: Looking into Personal Service Composition

Augusto Celentano, Stefano Faralli, Fabio Pittarello

Università Ca' Foscari di Venezia, Dipartimento di Informatica
Via Torino 155, 30172 Mestre (VE), Italy
(auce,faralli,pitt)@dsi.unive.it

Abstract. In this paper we discuss about composing services from standard suites of applications for personal data management in mobile devices. We propose a data model and an interface model that allows users to navigate among services according to the situation evolution 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 composer of personal services.

1 Introduction

Service composition is usually approached as a problem to be solved automatically: 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 [1] play a big role in such a view: the needed services are related to a user state that can vary only within defined boundaries, which do not influence his/her activities (hence his/her 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 [2]: 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 collected. In practice, users start 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 [3].

In this paper we discuss about composing personal activities supported by information stored in portable devices like UMPC, PDA, smartphone, etc. Such devices offer standard suites of applications for personal data management: agenda, address book, mailing and messaging, maps, etc., often independent or loosely coupled, used with different skills according to the personal style of the user. Such applications compel the user to find own way to compose and integrate the services offered by the device by 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.

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. 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 composer of personal services.

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. The word *situation* is replacing the word *context* in today literature [4], moving from physical, observable features of the user environment to a higher level of context including the user goals, plans, activity and history [5]. Situated computing derives from Situated Cognition [2], 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 [1], 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 [3, 6].

Situated computing is strongly related to the concept of adaptation of information and interaction, and exploits in mobile computing its full potential, even if its importance is not restricted to a continuously changing context. Context modelling, adaptation and interaction are the three keywords denoting the ability to design systems able to perceive the user status, to provide proper information and services, and to offer a user oriented effective interaction style.

In information processing applications, namely in the area of personal databases, closer 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 [7–9]. 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 in [10–12]. As noted in Section 1, their primary focus is to automate service discovery and composition, while our proposal puts the user in the center of the service composition choices.

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. Zoomable user interfaces (ZUIs) and gestures improve the usability of small devices applications [13, 14]. Zooming technology is becoming popular in new portable systems, like Microsoft WM 6.0-based devices and the Apple iPhone/iPod. SDKs and third part APIs, 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 framework with custom provided components. The applications built with such interfaces, while sharing with our proposal the smooth transition between overview and detail view on user data and services, do not help the user to change the perspective while browsing data. According to our vision, the system should allow the user to select information and services belonging to a category and present, on a synthetic view, the related items belonging to other categories. Each item should be accessible in a detail view or used as the focus of a new perspective in a highly dynamic fashion.

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 are 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 more the plan will survive to new information. 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 scarcely *queried* but 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.

This type of user behaviour 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 overlapping elements (e.g., a specific location shared by the current activity and by other activities) may become relevant if the situation changes; such elements, if properly evidenced, may act as switching points that can suggest the user to begin another activity.

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.

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.

A system supporting personal activities should:

1. Know about user activities and tasks, 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 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 an execution order, and by filling variable data with information taken from the 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.

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, a mapping between the schema and the personal information management system, and an interface able to explore the information and the services contributing to the user situation, zooming into the details, covering different areas of the personal data, thus supporting the user in the personal composition of services.

4 The Situation Model

The model of activities and services on which the situation lens is grounded is close to the *situation metaphor* defined by Hewagamage and Hirakawa [5]. It is built around four classes of objects: activities, tasks, events and services. In the following, details are omitted where not relevant in the scope of this paper.

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. The initial state represents the user needs as defined at activity planning, and the set of tasks is both a plan for the activity and its history, when tasks are done. The initial state of an activity can be empty, semantically representing a placeholder for an activity anticipated by the user but not planned at all.

Task. A task is a tuple $t = \langle task\_name, description, ES, status \rangle$. *Task_name* and *description* have obvious meanings; ES is a set of pairs $\langle e, s \rangle$, meaning that upon occurrence of event e service s can be executed; *status* is an enumerative value (*to_do*, *in_progess*, *done*) that can be set by service execution.

Event. An event is a tuple $e = \langle event\_name, time, location, contact, message \rangle$. An event can occur at a specified time, during a time interval, when reaching a location, upon a message arrival, or can be associated to a contact, e.g., adding a new contact triggers the event.

Service. A service is a tuple $s = \langle service\_name, description, A \rangle$, where A is a set of actions: *execute_service*, *create_event* and *change_status*, with obvious meaning. 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, e.g., in the shape of an URL fed to a browser.

Situation. The data instances describe a state of the user activity. Instances taken at different time stamps 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 Hewagamage and Hirakawa model [5].

It is worth to note that in the above model events and services are prototypes to be refined on the actual PDA software environment. In the definition of an event, time and location 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

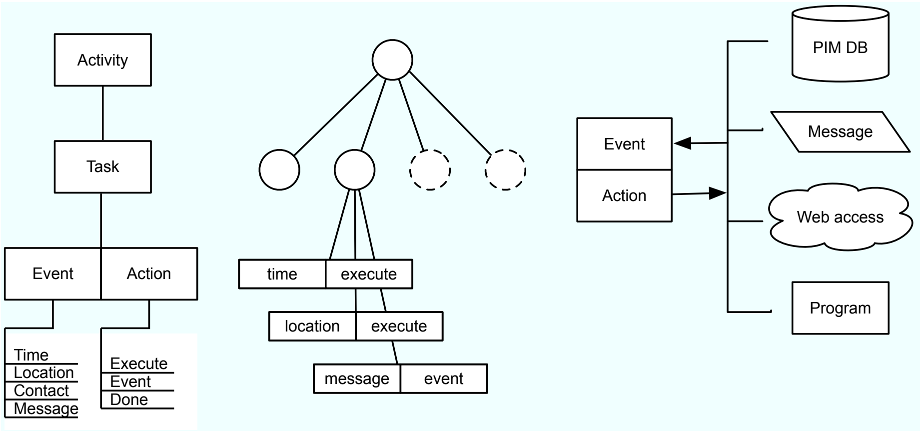

Fig. 1. The situation lens model

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 PIM Database 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.

5 The situation lens

The *situation lens* is a dynamic filter on the data managed by the 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;
- 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 an index to the task details;
- a *detail perspective*, allowing the user to explore detail data extracted from the personal information database and to execute programs and services.

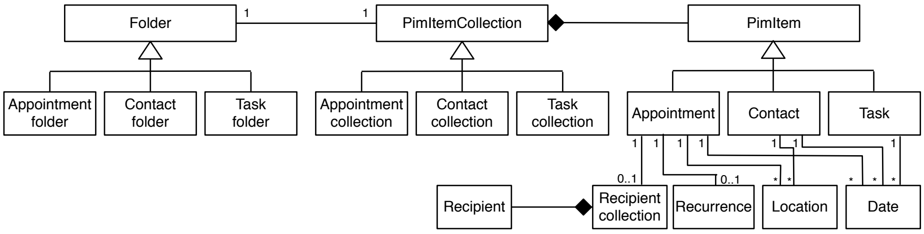

Fig. 2. The Pocket Outlook Object Model

In the following of this paper we assume as the underlying platform the Windows Mobile 5 system. The discussion is however general and largely independent from the platform.

5.1 Mapping the situation to the services

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 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 it.

Figure 2 shows a fragment of the *Pocket Outlook Object Model* (POOM), implemented in Windows Mobile systems. 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 similar (e.g., adding, removing and sorting items), the relations among data 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).

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 the information corresponding to keys to contacts, appointments, actions, messages, and possibly other information like maps. Figure 3 shows an example of the *task* and *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

Fig. 3. The task and detail perspective interface

Figure 3 shows the information of one of the two contacts of the task, while the right screenshot shows a map centered on the location associated to the task. The PDA services activated show their own interface and behavior, as if opened directly from the *Program* menu of the PDA.

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. Generic services can be activated from the PDA program menu.

5.2 The situation lens interface

Figure 4 illustrates the interface of the situation lens *global perspective*, giving 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.

The four screenshots show different views on an activity or a task. The first screenshot shows the interface structure. The situation lens 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 day to year, whose selection coherently updates the four quadrants' content.

The second screenshot shows the lens focused on an instance of a task; the number of icons in the other quadrants matches the number of instances of locations, services and contacts for that task. Icon variants describe different categories of instances (e.g., contact groups and service types). Locations are marked with different levels of grey to show the distance from the user location, sensed by a position sensor or assumed as a static information from the PIM database.

Fig. 4. The situation lens global perspective interface

Tapping and holding on an icon shows identification labels (names) and a menu for information filtering and direct access to the detail view. Tapping on an icon changes the situation lens focus to the icon. The third screenshot shows the lens focused on a person, displaying the tasks, locations and services linked to that person. A menu item associated to each icon allows the user to relate instances of the entities, as shown in the fourth screenshot. Keeping the situation lens focus on the person, the selection of a specific task highlights the subset of locations and services displayed on the screen that are associated to that task.

6 Conclusion

We have implemented a partial prototype in the Windows Mobile Device Simulator environment of Microsoft Visual Studio, to verify the feasibility of the approach both from the data management and from the user interaction points of view.

The implementation is at an early stage: data access and services interconnection have been tested, but the global perspective interface is a stand-alone module, allowing the user to explore the relations among the entities by "rotating" the lens, but acting on a set of data simulated with internal data structures.

The use of the simulator on as desktop PC lessens also the perception of the interface usability, due to the different gestures which distinguish mouse based and pen based operations. Also, the new generation of multitouch devices like the Apple iPod and iPhone will open new possibilities for natural gesturing. Nevertheless, the early experiments show that switching between entities 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.

Compared to the situated metaphor of Hewagamage and Hirakawa [5], 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.

Further work will be devoted to implement and to integrate the situation lens interface on Windows Mobile systems and to test it in a number of professional environments, starting from the academic and research worlds whose features match the scenario we have described.

References

1. Bertelsen, O., Bodker, S.: Activity theory. In Carroll, J., ed.: HCI Models, Theories, and Frameworks: Toward a Multidisciplinary Science. Morgan Kaufmann, San Francisco (2003) 291–324
2. Clancey, W.J.: Situated cognition: on human knowledge and computer representations. Cambridge University Press (1997)
3. Suchman, L.: Plans and Situated Actions. Cambridge University Press, New York (1987)
4. Dourish, P.: A foundational framework for situated computing. Position paper for the CHI 2000 Workshop on Situated Computing: A Research Agenda (2000)
5. Hewagamage, K.P., Hirakawa, M.: Situated computing: A paradigm to enhance the mobile user's interaction. In Chang, S.K., ed.: Handbook of Software Engineering and Knowledge Engineering. World Scientific (2000)
6. Bardram, J.: Plans as situated action: An activity theory approach to workflow systems. In: ECSCW. (1997)
7. Bolchini, C., Quintarelli, E.: Filtering mobile data by means of context: A methodology. In: Proc. OTM Workshops 2006 - Context-Aware Mobile Systems. (2006) 1986–1995
8. Bolchini, C., Schreiber, F.A., Tanca, L.: A methodology for very small database design. Information Systems **32** (2007) 61–82
9. Tanca, L.: Context-based data tailoring for mobile users. In: Proc. BTW Workshops 2007. (2007) 282295
10. Berhe, G., Brunie, L., Pierson, J.M.: Distributed content adaptation for pervasive systems. In: ITCC 2005, International Symposium on Information Technology: Coding and Computing, Las Vegas, Nevada, USA (2005) 234–241
11. Costa, P.D., Pires, L.F., van Sinderen, M.: Architectural patterns for context-aware services platforms. In: Proc. of the 2nd International Workshop on Ubiquitous Computing, IWUC 2005, Miami, FL (2005) 3–18
12. Strang, T., Linnhoff-Popien, C.: 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 (2003)
13. Bederson, B.B., Clamage, A., Czerwinski, M.P., Robertson, G.G.: DateLens: A fisheye calendar interface for PDAs. ACM Trans. Comput.-Hum. Interact. **11** (2004) 90–119
14. Karlson, A., Bederson, B., SanGiovanni, J.: AppLens and LaunchTile: Two designs for one-handed thumb use on small devices. In: Proc. of SIGCHI Conference on Human Factors in Computing Systems. (2005) 201–210