Enabling Informed Decision Making through Mobile Technologies: A Challenge for Software Engineering

Xavier Franch[1], Anna Perini[2] and Norbert Seyff[3]

[1]Universitat Politècnica de Catalunya (UPC), Spain
[2]Fondazione Bruno Kessler (FBK), Italy
[3]University of Zurich (UZH), Switzerland

franch@essi.upc.edu, perini@fbk.eu, seyff@ifi.uzh.ch

Abstract. The potential of mobile technologies is not fully exploited by current software services. One of the most influencing reasons for this problem is the lack of novel software engineering methods and tools that can master the complexity of mobile environments. Looking at a person in a smart environment, where mobile technologies and sensors are installed to support daily activities, it is observed that informed decision-making with the help of mobile technologies is beyond what users can expect from current software services. In this paper we present a motivating scenario to highlight the limitations of current decision support approaches. Based on this discussion we identify significant software engineering challenges, which currently hinder the realization of advanced decision support. In our research we have developed an initial version of a comprehensive framework that allows overcoming the challenges identified. It furthermore highlights which software engineering research lines may help to realize this vision.

Keywords: Software engineering, software services, apps, mobile platforms, service-oriented computing, personal cloud, decision-making, thoughtful living.

1 Introduction

The number and variety of software services (e.g., web services, mobile apps) dramatically increases every year. Service providers continuously emerge and the portfolio they offer grows steadily. Mobile technologies provide access to these services and are therefore becoming ubiquitous in our society. This leads to a magnitude of growth that was hardly conceivable in the recent past (e.g., the number of mobile phone subscriptions reached 5.000 million in 2010). Furthermore, it opens a lot of unforeseen opportunities for citizens worldwide and has improved citizenship's quality of life (West 2012).

We are particularly interested in supporting informed decision-making with novel mobile applications and services. We envision that such services can further improve individual citizen's quality of life and will also lead to more thoughtful use of resources and therefore thoughtful living of citizens. However, this vision currently goes beyond state of the art software engineering techniques and approaches.

In order to realize our vision we have identified promising work in areas such as context-awareness, personalization and evolution of services. In this paper, we present a proposal to improve existing work in these areas to boost the impact of current software service technologies at the individual and the society level. For awareness, we propose to include knowledge about the individual and about the environment in the heart of mobile technologies. For personalization, we propose (semi-)automatic orchestration and enactment of software services according to a user's past behaviour. For evolution, we propose that it is driven by needs of individual citizens rather than developer assumptions.

We foresee that achieving these goals in the near future is plausible due to the significant and continuous advances in mobile technologies. However, in our opinion software engineering methods and tools are lacking behind the fast advances in mobile technologies. We have identified several challenges within the above-mentioned areas. Among them the fact that services nowadays lack a semantic layer and push their users to learn new rules which are imposed by their providers. This lack of standardization can demotivate potential platform users and contradicts with the interest of service providers to increase the usage rate of their services.

In this paper we also present our vision of a semantic service engineering framework, which could allow users to interact seamlessly with mobile technologies. Such an easy-to-use approach would encourage all different kinds of potential users to adopt the framework. Automatic service enactment would allow exploiting techniques from other fields, such as machine learning.

The rest of the paper is organized as follows (see Fig. 1). Section 2 presents a scenario highlighting today's decision-making approaches and Section 3 discusses issues with regard to the presented scenario. Section 4 highlights how decision-making support could look like in the future. In Section 5 we discuss software engineering challenges in order to achieve our vision. Section 6 provides a first solution idea by depicting the emerging ecosystem behind our vision and outlining a semantic platform supporting informed decision-making. Section 7 discusses the lines of research where advancement is needed to realize our vision. In Section 8 we highlight related work and Section 9 concludes the paper.

Fig. 1. Organization of the paper.

2 Motivating scenario

Katie is the head of the paediatrics surgery unit at the Feeling Better International Hospital in Barcelona. Every day, her unit works with more than 100 patients. This work includes standard treatments that require only 30 minutes of their time, but also complex surgery lasting for several hours and involving several doctors. Her unit includes 30 doctors, 40 nurses and 10 administrative staff members. Most of the doctors are also academic staff of the Medical School at the Barcelona University. This means that on top of their medical duties, they have teaching responsibilities and need to take care of research projects (which might involve travelling).

Therefore, it is normal that members of her team are active from early morning to late at night. Although a daily schedule is available, it has to be reorganized in many cases as unexpected events are occurring (e.g., an operation takes longer). Observing the everyday work of her team, Katie has learnt that when this happens doctors feel distracted and even might think about possible appointments they have to cancel or reschedule while performing surgery. Furthermore, working late causes that doctors are tired and stressed. This is also worsened by the fact that most staff members live outside the city and have to travel for more than an hours on the average. Therefore, Katie has set up a new policy. In case a doctor finishes work later than 8 pm, the hospital offers free accommodation for the night including the transportation to and from the selected hotel. Furthermore, the hospital offers the doctors to manage their agenda and to inform family and friends about re-scheduling and delays.

Although Katie was confident on the success of the initiative, she observed problems. Managing the transportation and accommodation issue was not trivial because a doctor has to finish the on-going task before he can be asked about his preferences, therefore: 1) secretarial support staff complained about staying longer to take care of this service, 2) the doctor had to wait for the service, 3) different approaches to make a booking caused further delays, 4) spontaneous booking of a room or transportation was problematic, and 5) some doctors rejected to use the provided agenda management services, as they did not want to provide open access to their personal calendar. Katie concluded that a different solution was needed.

3 Analysis of the current scenario

The scenario above presents some issues that make the current support for doctors unsatisfactory:

Individuality. Every doctor is an individual with very different preferences, abilities, resources, etc. A one-fits-all solution might not be applicable. Katie is aware of this and, thus, she would like to offer services negotiated on an individual basis.

Privacy. Doctors are reluctant to make their private agenda public at the level required by the novel services offered. They do not want hospital staff to know about their private appointments and only share this information with the other parties involved and (possibly) their family. Therefore, Katie cannot have all the information needed to make the best possible decisions.

Service heterogeneity. Different hotels use different booking strategies which complicates the booking. Secretaries first have to identify the suitable booking procedure and often have to perform time consuming activities while booking (e.g., re-entering personal information of a doctor).

Lack of information. The information on which hotels and transportation options are still available for that day is not upfront available to the secretarial staff. They often have to fill in request forms to later find out that no more room is available.

Agility. As a consequence, the envisioned processes are not as agile and flexible as Katie would like them to be. Furthermore, the current solution often results in loosing time and requires additional resources, which negatively affects the hospital.

It is worth to mention that other scenarios could eventually reveal similar problems. For instance, doctors use to work in teams that may be preconfigured (e.g., same speciality) or dynamic (e.g., for a particular surgery). Sharing documents and information through hospital-centric applications may seem useful at a first sight, but may raise similar problems (e.g., privacy of confidential patient data records).

These limitations make Katie wonder about the possibility of alternative scenarios that are able to better exploit current mobile technologies.

4 Envisaged scenario

Katie consults the software engineering research team from the local university to find out how mobile technologies could support her in finding a solution. The researchers highlight that one possibility to achieve her goal could be to shift the focus from a central, hospital-based perspective, to a distributed, person-based point of view where doctors themselves are the ones who have full responsibility. Provided services would then fit with their own individual needs (also regarding privacy). Together with the researchers, Katie discusses a scenario where personal mobile devices suggest actions to doctors, or even execute them on their behalf. This approach avoids the assignment of new tasks to the hospital administrative staff, and simultaneously simplifies doctors' daily life. Austin, who is one of the most prestigious surgeons in the Feeling Better International Hospital, is the key person within this scenario. He is young, ambitious and loves his job, so he often accepts a certain overload in his daily work. On a particular day, he was expecting to finish at 7.30 pm but an unexpected problem with medical supplies has postponed the start of the last operation of the day (Norman's cardio-surgery) from 5 pm to 8 pm. Katie offered him to delay the operation until tomorrow, but the next day Austin is flying to Brussels for a project meeting early in the morning, so he decided to go ahead.

Luckily, he recently bought a smartphone with access to a novel platform supporting informed decision-making. This smartphone offers a lot of capabilities whilst being quite simple to use. It reacts to changes in the agenda and reschedules appointments accordingly. The following activities happen:

- Katie reschedules Norman's cardio-surgery in the hospital information system to start at 8 pm. This change is propagated to Austin's personal agenda.
- Two events are still scheduled in Austin agenda for after the operation. The first one is "buying a present for his mother's birthday next week". The platform just reallocates this task to another possible day before the birthday.
- The second event is different, a romantic dinner with his friend Angie at 9.30 pm. Since the operation is expected to last 2.5 hours, the platform knows that it has to cancel this appointment (differently from above, the event cannot be rescheduled without interacting with the interested parties). The platform sends Angie a nice apology message, specially designed by Austin in advance.
- The platform also detects the early morning flight to Brussels (leaving at 6 am). Considering Austin's travelling record track, the platform decides to book Austin a room in a hotel near the airport. Since the platform knows that, unless otherwise stated, Austin always drives his own car to the hospital, no taxi is needed.
- Finally, the platform sends Austin an e-mail with the summary of actions. This also includes a booking reference for the hotel and the parking space at the airport.

Once Austin leaves the operating room, he checks his smartphone and reads those messages. He feels reassured that his new device works correctly. He remembers, that after he started to use the novel platform he needed some time to get familiar with the system and also the idea that the platform has access to all his personal data. With the current level of configuration and the history available, he is more than happy with the way it behaves.

Not only Austin is impressed by the mobile platform, also Katie can see that this new approach allows overcoming the issues identified in Section 3:

- No other person (e.g., secretary) is needed to take care of individual user preferences. The user-centric system is able to tailor itself to the individual needs of a person. The mobile platform knows a user’s preferences and analyses the decisions taken by a user. This allows the platform to make suggestions and to handle situations intelligently, which makes it the ideal companion.
- Advanced privacy mechanisms give Austin full control about privacy relevant data and letting him actively decide which information is shared and with whom. Furthermore, the device automatically identifies certain levels of trust based on Austin’s past sharing activities, which it follows when sharing information on Austin’s behalf.
- Different services and technical details, this all is transparent for Austin as a user. The new platform takes care of identifying relevant services and presents information in an easy to understand and personalized way.
- This also means that there is no lack of information any longer. As everything is automated, the mobile platform can highlight the availability of rooms and transportation options in real time.
- This finally leads to the envisioned agile process and gives doctors the needed freedom, so that they can focus on work. No additional resources are needed and a certain improvement can be obtained.

5 Software engineering challenges

In the following we highlight key challenges towards strengthening user-centrality and enabling a user-driven evolution of software applications. Overcoming these challenges would allow implementing the scenario described in Section 4. The relationship between these challenges and the open issues mentioned in Section 3 is visualized in Fig. 2.

5.1 Strengthening user centrality

Modern mobile devices, such as smartphones, are equipped with numerous sensors. However, approaches that allow determining a user’s context are still limited. Furthermore, services still require the end-user to make a tedious personalization job. The support given by providers to adapt services to the needs of an individual user is quite limited and just includes some basic characteristics (e.g., language selection). These issues limit current services to react based on the given environment and particular user profiles and needs. In our scenario the hotel booking action is an example. Selecting a hotel that fits best may depend on the place of the first morning commitment. As Austin needs to be ready for his 8 am class on a Tuesday, the system will book a room near the university. Current software engineering approaches and tools do not sufficiently support such considerations and the following challenges have been identified:

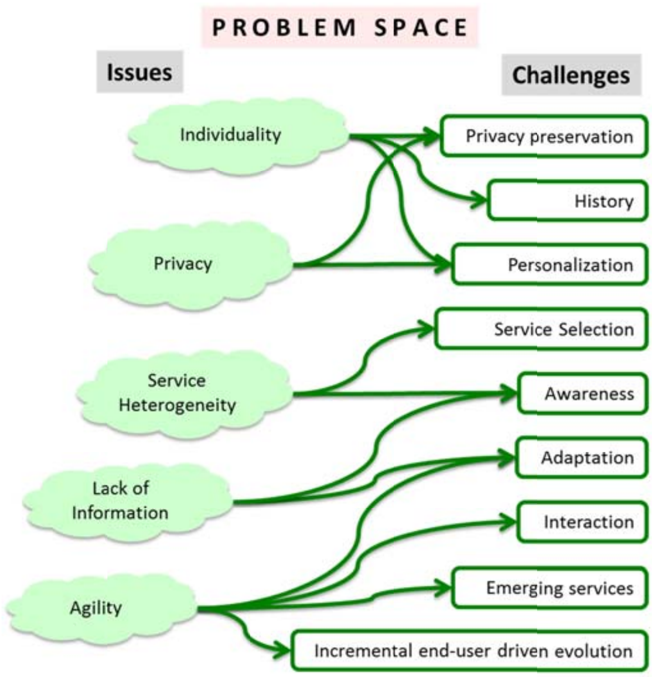

Fig. 2. Issues and associated challenges.

Privacy preservation. Success of an IT product heavily relies on respecting privacy. Sensible data must be kept inside the individual's boundary. For example, Austin prefers to keep his friendship with Angie confidential. The communication of information needs to be balanced. While some kind of aggregated, anonymous data sent to the service providers may help them to analyse service usage and improve their products in the future, private data needs to stay under user control.

History. Individuals tend to apply patterns of behaviour. These patterns may be tacit and usually will emerge after some time. For instance, being young and energetic, Austin does not mind sleeping in hotels paid by the hospital, whilst other doctors (e.g. with children) will prefer to sleep at home and take a taxi to the airport early in morning. Their decisions along time will reflect their preferences. This means that decisions need to be monitored and analysed in order to support decision-making in the future.

Personalization. A selected service furthermore needs to be tailor to individual customer preferences. This for example includes how the service presents its functionalities to the user (the look and feel of the service), but personalization also needs to consider the data that the user is willing to provide as input for the service.

Service selection. Several services might be available to support a selected task. The service providing the best value needs to be selected. This does not only mean to select the service providing the best performance, but also to consider the costs of a service.

Awareness. Decision-making may be improved by increasing the awareness about the environment. For instance, the platform could be informed about the current location of Austin's car. This could improve decision-making so that the need of calling a taxi for bringing him to the airport is automatically identified.

Adaptation. The increasing availability of information paves the road for better decision-making. The best decision today may not be the best tomorrow. For instance, a hotel, which now has good ratings, might be a bad choice next year. Furthermore, sudden adaptation is required as unexpected events might occur (e.g., bad weather forecast in Brussels) may require an unexpected reaction (e.g., Austin going back to his house for taking his coat). This means that services need to adapt constantly to a changing environment.

Interaction. As part of the individualization aspect, some citizens may rely more than others on technology. Whilst Austin seems to be fully confident in the mobile platform, other doctors who have similar smartphones may choose the "Always Require Confirmation" option. Therefore, we need to be aware of and respect different ways of interacting with a possible solution.

5.2 Enabling user-driven evolution

Literature highlights that software must be adapted and enhanced continuously to remain satisfactory (Bennett and Rajlich, 2000). User needs and expectations change over time and services should provide the desired new features. Furthermore, they need to improve in quality. Currently, methods and approaches to identify changing user needs are limited and do not allow to continuously involve end-users in service evolution.

Emerging services. The dramatic increment of available apps and services requires improved mechanisms to identify interesting functionalities that emerge from providers of any kind. For instance, Austin should be offered new services on transportation. In case Austin is interested to try these services should be automatically integrated into the workflow required. Filtering, recommendations and crowdsourcing become cornerstones of this idea. Individuals may play a part in this scene by publishing (i.e., sharing) their own services thus actively contributing to the service marketplace population.

Incremental end-user driven evolution. Functionality provided by a service platform may grow by increments, as an average user needs some time to master a new service. Then new needs may be identified. For instance, Austin may at some time investigate how to use the invoice generation facilities by hotels in order to store a copy of such an invoice in his Dropbox account for his own purposes or post this need to developers if not yet provided.

In the next section we propose a high-level architecture that aims to overcome these challenges.

6 A platform for semantic service engineering

The envisioned scenario can be generalized in terms of an emerging ecosystem. Citizen using smart devices are the key component. We envision that the mobile device can make suggestion to the user based on analysing existing information. Information can be gathered via the smart environment (environmental data) or services and apps. Communication with other systems and users can furthermore include relevant information. Results of the analysis of the gathered information might also be communicated to other systems and users. A particular example is the communication of feedback to developers in order to ensure continuous service evolution. Fig. 3 depicts this ecosystem. Given their current predominance in society, smart mobile devices (smartphones, tablets, etc.) provide all necessary functionalities to help people, e.g. citizens of a smart city, organizing their daily activities and accessing a variety of services through apps. We envision that over time, the appropriate apps are discovered and installed according to the citizen's profile. This profile resides in the citizen's personal cloud that contains all sensible information that needs to be private. Being in the cloud, this information is shared by all mobile devices used by the citizen, thus preventing problems in synchronization of data and profiles. The profile goes beyond the typical concept that is applied for using applications today, we envision a social profile that emerges from past actions and feedback given by the citizen to suggestions that the mobile devices provide over time, which can be analysed by developers and motivate them to evolve the service or app it refers to. The mobile device is tightly connected to the environment, especially to the smart city that surrounds the citizen, and any possible sensor that the citizen may use (e.g., smart clothes for medical monitoring). With all this information and also interoperating with more classical information systems that are of interest for the citizen (e.g., at work), the mobile device may take decisions on the go and inform other individuals about the consequences of these decisions.

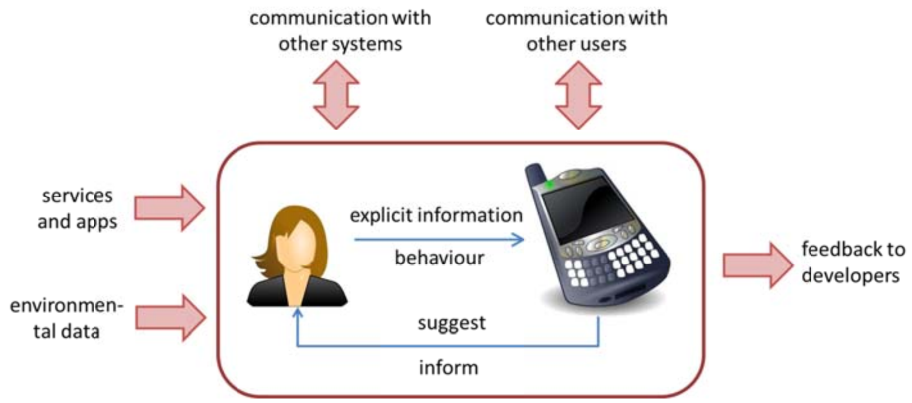

Fig. 3. The emerging ecosystem.

Focusing on the logical architectures of the software platform that can enable the described ecosystem, and allow addressing the previously discussed challenges, we can first recognize that the choice of whether to put a decision-making component on-board of the mobile device or in the personal cloud or distributed on both, brings to a family of possible architectures. Fig. 3 shows one of these possible architectures, adopting distributed decision-making (i.e., the decision-making component is embedded in the mobile device). Its three main logical components are presented next.

Table 1 at the end of the section shows how the different mentioned platform elements are involved in the software engineering challenges identified in Section 5.

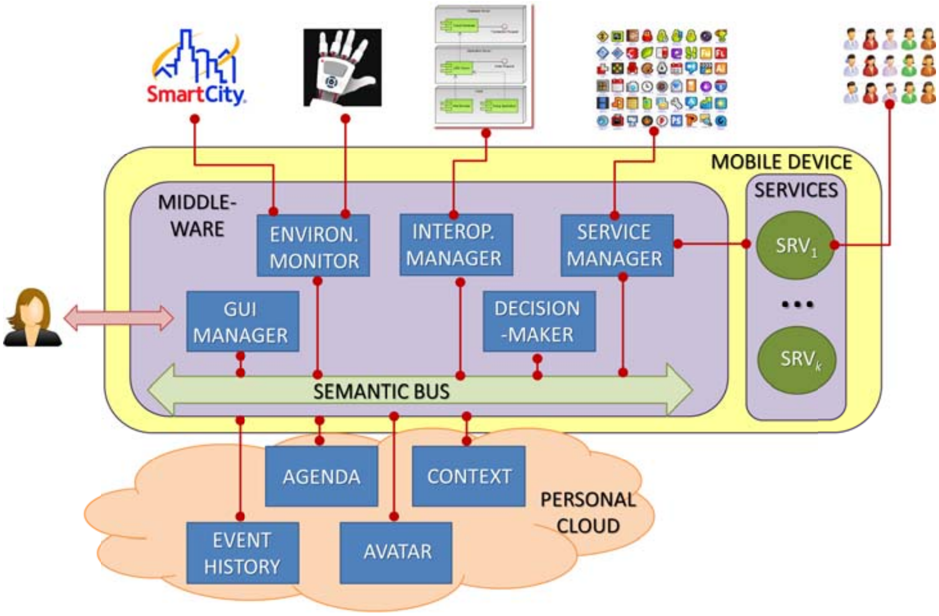

Fig. 4. High-level architecture of a platform for enabling informed decision-making.

6.1 The personal cloud

Within the personal cloud component we identify four main resources, each controlled by a specific manager (not shown in the figure):

- *Agenda.* Keeps track of the citizen's daily activities. This agenda needs to be seen as multidimensional, in which the usual time arrangement is just one possible viewpoint; others like location, people met, etc., should be easily retrieved in order to facilitate later management.
- *Context.* Represents the context of the user, continuously updated. Under this context, we may identify the following dimensions: time, location, environment (e.g., low battery), type of activity (e.g., work, leisure, family), user skills (novice, experienced; may be application-dependant), etc.
- *Event history.* Stores the activities of the citizen in the past. In the general case, activities, even if represented in the form of a list (time-ordered), encode complex workflows that represent processes followed by the user.
- *Avatar.* Creates a representation of the citizen that is used for decision-making. This can be done exploiting decision-making algorithms executing in the personal cloud, which, for instance, dynamically rank alternatives (e.g. services, or products) along the user's preferences.

6.2 The mobile device

The *mobile device* includes the following elements:

- *GUI manager.* Intelligent interface of the mobile device with the user, supporting agile composition and personalization.

- *Decision-maker.* This is the real core of the platform. It continuously decides about the next actions to make.
- *Environmental monitor and interoperability manager.* Communicate with the outside world (smart cities, body sensors, information systems, etc.) and also get the relevant environmental information from the mobile device (battery level, etc.).
- *Service manager.* Discovers and, when appropriate, installs services in the mobile device. This installation includes an initial automatic personalization.
- *Service space.* Set of services installed on the device, including those modified or defined by the user. Some of them may be used to inform other users affected by decision-making, some others to provide explicit or implicit (i.e., logs of service/app usage) feedback to service and app developers.

6.3 The semantic bus

The semantic bus component is a classical interoperability bus for event-driven communication, with the particularity that events have a high semantic content. It implements a publisher-observer pattern.

Table 1. Relating the solution space with the problem space

		SOLUTION SPACE										
		Personal Cloud				Mobile device						
		Agenda	Context	Event history	Avatar	GUI manager	Decision maker	Environ. monitor	Service manager	Service space	Semantic bus
PROBLEM SPACE	Privacy preservation	+			+		+			+	+
	History	+		+			+		+		
	Personalization	+		+	+	+	+		+		
	Service selection			+	+		+	+	+	+	
	Awareness	+	+				+	+	+		
	Adaptation	+	+	+			+		+		
	Interaction				+	+					
	Emerging services	+	+	+			+		+	+	+
	Incremental evolution			+			+		+		+

7 The way ahead

The realization and adoption of the envisaged platform, and the need to enable seamless evolution of services and applications in the emerging ecosystem, ask for advances in software engineering research, which may also take advantages of results and techniques from different research fields. In this section we enumerate emerging synergies and promising research lines as a preliminary step towards setting up a research agenda.

Semantic interoperability. In order to allow interoperability among all the platform components and the external services, ontologies are needed to represent the information that flows around according to some agreed conceptual reference framework (Uschold and Gruninger, 1996). General ontologies for time, localization, etc., from organizations like W3C, could be adopted to serve as lingua-franca for the platform. Data produced and consumed by services should be compliant to these ontologies in order to allow interconnection through the platform.

Knowledge representation and reasoning techniques. In order to build an accurate and trustable knowledge base and infer the behaviour that better matches users' expectations, AI techniques for knowledge representation and automated reasoning are promising. Recommender systems (Adomavicius and Tuzhilin, 2005) may provide (even automatically execute) recommendations on which services to apply; some applications in the marketing context, e.g. (de Bruin et al., 2008), have explored this particular aspect. Case-based reasoning (Aamodt and Plaza, 1994) may be useful to improve the knowledge and reasoning capabilities of individual users case by case. More recently, the use of ontology in combination with statistical models is proposed to provide models of human behaviours in a given context (Codescu, et al., 2011).

Service solutions. A great deal of existing proposals in the service-oriented computing field clearly transfers into our envisaged platform. Approaches for service discovery (Ran, 2003), service composition (Rao and Su, 2005) (in particular using AI techniques (Beauche and Poizat, 2008)) and service adaptation (di Nitto et al., 2008) are of application to satisfy some of the envisaged challenges.

Social collaboration. The active participation of a large number of people to perform particular tasks or solve problems, as emerging in the so called social computing (Wang et al., 2007), is of great interest for our work, and has been pointed out as an opportunity to exploit in different software engineering processes, from requirements engineering to software testing. Lim and Finkelstein (2012) have investigated first approaches towards large-scale requirements elicitation using social networks. These approaches complement classical market-driven requirements elicitation methods (Karlsson et al., 2007). The potential of social collaboration (via social network platform), to tackle the issue of "unknown unknown" requirements is pointed out also in (Sutcliffe and Sawyer, 2013). Crowdsourcing for addressing the Oracle problem in software testing has been also recently investigated (Pastore et al., 2013). Furthermore, Onnela and Reed-Tsochase (2010) provide first insights on social influence within social networks.

Participatory sensing. Gathering contextual information in order to allow services to adapt to a particular user context is a key aim of our work. Research on participatory sensing (Burke et al., 2006) focuses on communities that use sensors as provided by mobile devices to retrieve information about the environment.

adfa, p. 11, 2011.

Change Management. Different sources of change need to be identified, classified and analysed. Research on self-adaptive software services focus on defining solutions to managing dynamic changes in the environment, mainly by adopting a monitor-eval-adapt control loop (Di Nitto et al., 2008). Most of the proposed approaches focus on design-time methods, while more recent work aim at equipping service-based systems with mechanisms for managing changes of user requirements and preferences at run-time, e.g. (Qureshi et al., 2010). In this light, also traditional software evolution approaches require rethinking. Service providers need to be aware not just of new needs coming from the potential customers but also new opportunities coming from other services and applications. To this end, very agile change management processes need to be designed. The concept of "fluidity of design" (Jarke et al., 2011) should be accommodated somehow in these processes. One crucial question here is timing: when is the right moment to update the service, for which selected requirements?

Personal and social values. Beyond pure technological knowledge, personal and social factors need to be considered in this kind of solutions. Long ago Goguen (1994) already recognized this link in requirements engineering. The key value of requirements in this context was also recently highlighted by Milne and Maiden (2011) who demonstrated that requirements are socially constructed in a political context. This means that decision-making needs to consider all type of factors surrounding individuals. Towards this objective, studies in the area of social science, which are based on empirical survey techniques conducted on large user population can contribute with statistically relevant data about the relationship between lifestyle traits, social influence, people's attitudes towards mobile innovations and the adoption of various types of mobile services (Bouwman et al., 2012). Indeed, our vision includes the idea that people should benefit individually from the proposed solution, in harmony with the goals of a sustainable society, such thoughtful use of resources and energy.

Feedback/Communication channels. Continuous feedback on services is needed in order to ensure long-term user satisfaction. Approaches which allow end-user to give feedback on current context-aware services (Schneider et al., 2010) and which allow them to document their ideas on services in situ (Seyff et al., 2010) build a basis to satisfy some of the depicted issues.

Process mining. The fact that the proposed platform includes an activity history in the form of list makes process mining (van der Aalst, 2011) an interesting research field. In other words, the activity history may be considered as a personal process log in which existing techniques may be assessed and applied if adequate.

Table 2 sketches an overview on the relationships we can see so far between these research lines and the challenges, and solution elements above discussed.

8 Related work

In this paper we have proposed a novel approach to close the gap among regular citizens and software services available in mobile technologies. We aim at simplifying the interaction of multiple internet services by means of a dedicated platform that is able to make decisions autonomously and also to learn from past decisions from the user. Our vision relies on several existing works both in the form of scientific contributions and existing technologies that we survey below.

Table 2. Research lines vs. solution elements they can contribute for.

	SOLUTION SPACE		
RESEARCH LINES	Personal Cloud	Mobile device	Semantic bus
Semantic interoperability		+	+
Knowledge representation and reasoning	+	+	
Service solutions	+	+	
Social collaboration	+	+	
Participatory sensing		+	
Change management		+	
Personal and social values	+		
Feedback/Communication channels		+	+
Process mining	+	+	

The IFTTT Project (https://ifttt.com/) supports user-designed service composition. For example, a user can create a rule that is triggered when he uploads an image to Instagram that saves this image in his Dropbox account. Such rules (called recipes), can be shared among users or created in a personalised basis. A similar approach is followed by the SATIN project (http://www.satinproject.eu/mission). Although the system is not designed to learn from the user behavior, it opens the path to communication between applications. Such a technology could be integrated into the GUI manager with the purpose of supporting the user to configure his personalized workflows.

Similarly, the on{X} project (https://www.onx.ms/#!landingPage) lets the user control and extend the capabilities of his Android smartphone using a JavaScript API. on{X} provides an API that allows the device to detect several user events, as for example the speed of movement or the arrival to the office. Applications can use this API to react to these events. This type of technology can be integrated in the environment monitor e.g. to update doctors' context when they park the car at the hospital.

Also several applied research projects tackle related issues. The PERSIST project (http://www.ict-persist.eu/) envisions a Personal Smart Space (PSS) that is associated with the personal devices carried by the user and which follows him, providing uninterrupted context-aware pervasiveness. This concept of PSS could be the basis of the avatar component in our platform. The SOCITIES project (http://www.ict-societies.eu/) aims at improving on-line community services, creating new ways of communicating, working and socialising. In their own words, "the vision of SOCITIES is to develop a complete integrated Community Smart Space, which extends pervasive systems beyond the individual to dynamic communities of users".

MUSIC (http:// ist-music.berlios.de/site) developed an open framework for the development and deployment of context aware and self-adaptive mobile applications targeted for ubiquitous and service oriented environments. The framework offers a distributed context sensing and management system and supports self-adapting distributed mobile services collaborating in dynamically adapting ensembles. With the help of MUSIC, a developer can implement and deploy a custom context sensor specific to a given device (e.g. a sensor for handling compass data). Still, the framework requires significant effort and a case-by-case study to integrate new services into the user environment.

Some platforms start to be also available in mobile infrastructure. BLOCCO (Hagino et al., 2011) is a service linking system available in Android platform that enables the building of new application mash-ups by linking other existing Android applications. This was delivered in the form of an Android application. The main goal of the project was to enable users to combine functionalities provided by different applications and to implement automatic execution of applications according to user configuration. In addition, various events happening in one application could be detected and they could be used to trigger execution of other services, using parameter passing and processing techniques. Similar to IFTTT, BLOCCO focused on constant rules for end-user configuration and enabled end-users to build new applications according to their specific needs, in a user-centred fashion.

Finally, some academic works have already explored similar features or functionalities. An event-driven approach for business process modeling (Alexopoulou et al., 2008) was introduced to enhance agility by means of learning rules between events and actions. Similarly, the integration of adaptive process management and case handling was used to create a more flexible and user-friendly approach to process management (Gunther et al., 2008). Another noteworthy work (Mehandjiev et al., 2009) studies end-user service composition from the perspective of users. With this goal, the authors review users' perceptions, intuitions and requirements regarding bridging different services. Finally, Semantic Web Pipes (Le-Phuoc et al., 2009) is a mechanism that supports fast implementation of semantic data mash-ups while preserving abstraction, encapsulation, component-orientation, code re-usability and maintainability.

9 Conclusions

In this paper we have identified challenges for software engineering based on an envisioned example focusing on decision-making support of the future. Furthermore we have presented a first solution idea of a platform which could provide the envisioned decision support. However, in order to achieve this vision several advances regarding software engineering methods and tools are required. We discuss an initial research agenda where we have reflected on the different research lines that may contribute to the realization of our vision.

Whilst certainly there is a long path to achieve the scenario presented in Section 4, we have tried to show that a lot of work is already there and can be used as the baseline for building such a platform. Still, many fundamental questions need to be addressed. For instance, recent findings dispute the idea that people are rational decision-makers (Lehrer 2009). This opens an interesting debate: is it cost-effective to try to embody all possible preferences and attitudes of citizens in a computational form? Answers to this kind of fundamental questions allow us envisaging new emerging interdisciplinary research lines.

10 Acknowledgments

This work has been supported by the Spanish project TIN2010-19130-C02-01.

11 References

1. Aamodt, A., Plaza, E., 1994. Case-Based Reasoning: Foundational Issues, Methodological Variations, and System Approaches. *AI Communications* 7(1).
2. Adomavicius, G., Tuzhilin, A., 2005. Toward the next Generation of Recommender Systems: A Survey of the State-of-the-art and possible Extensions. *IEEE Transactions on Knowledge and Data Engineering*, 17(6).
3. Alexopoulou, N., Nikolaidou, M., Chamodrakas, Y. Martakos, D, 2008. Enabling on-the-fly Business Process Composition through an Event-based Approach. In HICSS 2008 – *Proceedings of 41st Hawaii International Conference on System Sciences*. IEEE Press.
4. Beauche, S., Poizat, P., 2008. Automated Service Composition with Adaptive Planning. In ICSOC 2008, *Proceedings of the 6th International Conference on Service-Oriented Computing*, Springer Verlag.
5. Bennett, K.H., Rajlich, V.T., 2000. Software Maintenance and Evolution: a Roadmap. In ICSE 2000 – *Proceedings of Conference on The Future of Software Engineering*, ACM Press.
6. Bouwman, H., López-Nicolás, C., Molina-Castillo, F.J., van Hattum, P.: Consumer lifestyles: alternative adoption patterns for advanced mobile services. *IJMC 10(2) (2012) 169–189*
7. Burke, J., Estrin, D., Hansen, M., Parker, A., Ramanathan, N, Reddy, S., Srivastava, M. B., 2006. Participatory sensing. In WSW 2006 – *Proceedings of Workshop on World-Sensor-Web: Mobile Device Centric Sensor Networks and Applications.*
8. Codescu, M., Horsinka, G., Kutz, O., Mossakowski, T. and.Rau, R., 2011 Osmonto - an ontology of openstreetmap tags. In *State of the map Europe (SOTM-EU)*.
9. Di Nitto, E., Ghezzi, C., Metzger, A., Papazoglou, M., Pohl, K., 2008. A Journey to highly dynamic, self-adaptive Service-based Applications. *Automated Software Engineering*, 15.
10. Goguen, J., 1994. Requirements Engineering as the Reconciliation of Social and Technical Issues. In Goguen, J., Jirotka, M. (eds.), *Requirements Engineering: Social and Technical Issues*, Academic Press Professional.
11. Gunther, C., Reichert, M., and van der Aalst, W, 2008. Supporting flexible processes with adaptive workflow and case handling. In *WETICE 2008 – Proceedings of Workshops on Enabling Technologies: Infrastructure for Collaborative Enterprises*. IEEE Press.
12. Hagino, H., Fujii, K., Murakami, J., Hara, M., 2011. The BLOCCO Service Linking System, Enabling Combination of Services through User Configuration, *NTT DOCOMO Technical Journal* 12(4).
13. Jarke, M., Loucopoulos, P., Lyytinen, K., Mylopoulos, J., Robinson, W., 2011: The Brave New World of Design Requirements. *Information Systems Journal* 36.
14. Karlsson, L., Dahlstedt, A.G., Regnell, B., och Dag, J.N., Persson, A., 2007. Requirements Engineering Challenges in Market-driven Software Development - An Interview Study with Practitioners. *Information and System Technologies* 49(6).
15. Lehrer, J., 2009. *How we Decide*. Houghton Mifflin Harcourt.
16. Le-Phuoc, D., Polleres, A., Hauswirth, M., Tummarello, G., Morbidoni, C. Rapid prototyping of Semantic Mash-ups through Semantic Web Pipes. In *WWW 2009 – Proceedings of International World Wide Web Conference 2009*. ACM Press.
17. Lim, S.L., Finkelstein A., 2012. StakeRare: Using Social Networks and Collaborative Filtering for Large-Scale Requirements Elicitation. *IEEE Transactions on Software Engineering,* 38(3).

18. Mehandjiev, N., Namoune, A., Wajid, U., Macaulay, L., Sutcliffe, A., 2009. End User Service Composition. In *WWW 2009 – Proceedings of International World Wide Web Conference 2009*. ACM Press.
19. Milne, A., Maiden, N., 2011. Power and Politics in Requirements Engineering: A Proposed Research Agenda. In RE 2011 – *Proceedings of the 19th IEEE International Requirements Engineering Conference*.
20. Onnela, J.-P., Reed-Tsochase, F., 2010. Spontaneous Emergence of Social Influence in Online Systems. *Proceedings of the National Academy of Sciences of the United States of America*, 107(43).
21. Pastore, F., Mariani, L, and Gordon F., 2013. CrowdOracles: Can the Crowd Solve the Oracle Problem. *International Conference on Software Testing, Verification and Validation (ICST).*
22. Qureshi, N. A., Perini, A., 2010 Requirements engineering for adaptive service based applications, *Proc. of the 18th IEEE Int. Requirements Engineering Conf.*, pp. 108–111.
23. Ran, S., 2003. A Model for Web Services Discovery With QoS. *ACM Sigecom exchanges*, 4(1).
24. Rao, J., Su, X., 2005. A Survey of Automated Web Service Composition Methods. *Semantic Web Services and Web Process Composition*, 3387.
25. Schneider, K., Meyer, S., Peters, M., Schliephacke, F., Mörschbach, J., Aguirre, L., 2010. Feedback in Context: Supporting the Evolution of IT-Ecosystems. In PROFES 2010 – *Proceedings International Conference on Product-Focused Software Development and Process Improvement.*
26. Seyff, N., Graf, F., Maiden, N.A.M., 2010. Using Mobile RE Tools to Give End-Users Their Own Voice. In RE 2010 – *Proceedings of the 19th International Requirements Engineering Conference.*
27. Sutcliffe, A., Sawyer, P., 2013. Requirements Elicitation: Towards the Unknown Unknowns. In *RE 2013 – Proceedings of the 21th International IEEE Requirements Engineering Conference*.
28. Uschold, M., Gruninger, M., 1996. Ontologies: Principles, Methods and Applications. *Knowledge Engineering Review* 11(2). Cambridge University Press.
29. van der Aalst, M. P., 2011. Process Mining - Discovery, Conformance and Enhancement of Business Processes. *Springer 2011, ISBN 978-3-642-19344-6, pp. I-XVI, 1-352*
30. Wang, F.-Y., Carley, K. M., Zeng, D., Mao, 2007. Social computing: From social informatics to social intelligence. *Intelligent Systems, IEEE 22.2 (2007): 79-83*.
31. West, D., 2012. How Mobile Devices are Transforming Healthcare. *Issues in Technology Innovation*, 18.