Towards a Framework for Intelligent Mobile Service Applications

FERNANDO KOCH

fkoch@acm.org
Intelligent Systems Group
Institute of Information and Computing Sciences
Utrecht University, The Netherlands

Abstract. This paper introduces the Framework for Intelligent Mobile Service, which defines the data structures and computational modules required to support the provisioning of intelligent mobile service solutions.
Keywords: Mobile services, artificial intelligence.

(Received May 27, 2005 / Accepted August 15, 2005)

1 Introduction

Intelligent Mobile Services are service applications that integrate mobile computing and service providers and provide the infrastructure to reason about the user's context, what enables the application to predict user's intents and actions [1][2]. This scenario involves complex systems that have multiple interactive components, immersed in a dynamic environment and limited resources. Moreover, intelligent behaviour involves the implementation of features such as pro-activeness, self-tuning, plan selection and context-sensitiveness. The implementation of these features brings together issues of distributed computing, mobile computing, personal assistance and artificial intelligence. The complexity of developing mobile service solutions is related to support the elements and interactions in this problem scenario.

To understand mobile service problem scenarios one must be able to describe the elements and interactions involved and the components required to support these elements. This work introduces an analysis of this domain of problem and defines the structures required to implement mobile service solutions. It introduces the elements and interactions in mobile service scenarios and analyses the components required to support these elements. Then, it integrates these components in the *Generic Architecture for Mobile Service Applications*. Finally, the study introduces the *Framework for Mobile Service Solutions*, which provides an overall picture of the elements and interactions involved.

This paper makes three main contributions to mobile computing literature. First, it introduces the description of the elements and interactions in the problem scenario, which provides a clear account of the components required to create intelligent mobile service solutions. In addition, it introduces the *Generic Architecture* that works as basis for the specification of future *implementation platforms*. Last, it introduces the *Framework for Mobile Service Solutions* that provides an overall picture of the elements and relationships in the problem scenario, and works as a guideline for the analysis and detail design phases.

This paper is organized as follows: Section 2 motivates the need for a framework that includes the elements, interactions and components of intelligent mobile service solutions and presents related work with early attempts to build these applications. Section 3 presents the elements and interactions in mobile service environments. Then, section 5 introduces the *Components* of intelligent mobile service applications and outlines the *Framework for Mobile Service Applications*. The paper concludes with section 7.

2 Motivation and Related Work

This section motivates the need for a method to describe the elements and interactions of intelligent mobile services solutions.

Mobile computing could be the tool to deliver *always present* personal assistance and provide user-centric solutions that aid the mobile user with information, collaboration and coordination [3]. The key technologies to implement these solutions involve the use of *context-awareness, collaboration, enhanced inference* and *device interface*.

Mobile service solutions are composed by a set of active and passive elements and the interactions between them. The solution to deliver intelligent mobile services involves the integration of distributed computing, mobile computing, personal assistance and artificial intelligence. Therefore, a reasonable question to

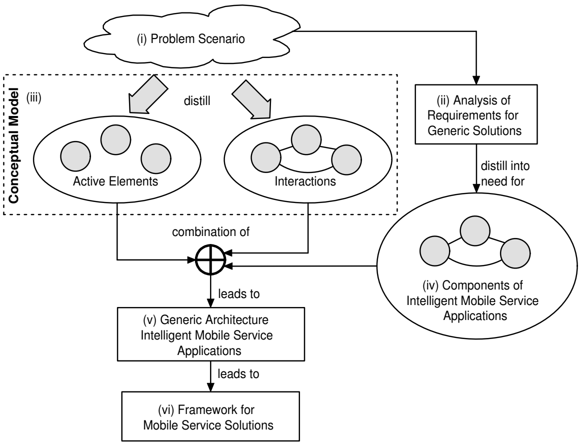

Figure 1: Towards a Framework for Mobile Service Solutions

ask is:

> *What is the set of components required to provision intelligent mobile service solutions?*

In this work, we propose a *Framework for Intelligent Mobile Services* that emerges from the analysis of the elements and interactions in mobile service problem scenarios and works as a guideline for the analysis and detail design phases.

Figure 1 presents the storyline of this study. First, we propose the analysis of an illustrative (i) problem scenario, which leads to the description of (ii) the functionalities required by a "generic" solution. The analysis of the problem scenario also introduces the (iii) active elements and interactions of the environment. The combination of these elements and interactions distils into the *Conceptual Model* of the problem scenario. As the analysis moves towards the detail design description, it introduces (iv) the *components* of mobile service applications, which are combined into (v) the *Generic Architecture for Mobile Service Applications*. The integration of applications, users and devices compose the overall picture of the mobile service problem scenario, describe in (vi) the *Framework for Mobile Service Solutions*.

2.1 Related works

Related works in the area include the ones aiming at delivering support to multi-user, mobile service interactions and personal mobile assistance. To our knowledge, there is no previous attempt to supply a formalized view of components in mobile service solutions and the the "development" standards follow the ones from software engineering adapted to those scenarios. For example, object-oriented programming is made available for the creation of mobile services through Java 2 Micro Edition [4] and other platforms. However, this development model targets the low-level application programming requirements leaving to the developer how to architecture the solution. The FIPA specification for nomadic applications support [5] deals with agent middleware to support applications in nomadic environment. This specification gives an overview of the *Nomadic Application Support* area and contains specifications for: Monitor Agent (MA) functionality, Control Agent (CA) functionality. However, in this level of abstraction this specification does not handle issues of mobile service solutions such as support to context-awareness, collaboration and device interface. The lack of formalized tools to define intelligent mobile service solutions leads to the adoption of ad hoc approaches, which come at the expense of generality and reuse.

For example, *MyCampus* [6] provides an agent-based environment for context-aware mobile services. The strong point in MyCampus is its openness through the Semantic Web representation of the context data. However, although MyCampus provides the infrastructure for data representation and collaboration, it does not provide a clear specification (or framework) for the development of the intelligent applications.

Alternatively, *Paurobally et al* [7] proposes an agent-based framework for providing personalised mobile services, with a view of producers and consumers of services as software agents, some of which are located on users' mobile devices. However, the focus is on the problem of negotiation (subset of collaboration), not supporting other aspects of intelligent mobile service provisioning.

On the other hand, *Electric Elves project* [8] exploits agent technology to assist users with daily tasks, such as organising meetings. The project includes the notion of *context-awareness* and *collaboration*. Although Electric Elves is a very complete project, it fails to consider a *generic* framework for the development of mobile service solutions.

From the analysis of related projects, it is possible to identify the lack of formal definitions for mobile service. In the next sections, this work introduces the elements of the problem scenario and proposes the diagram of the elements and interactions in mobile service solutions.

3 Mobile Service Problem Scenario

As mentioned, the problem of defining intelligent mobile services is related to define the necessary components to support the element and interactions in these environ-

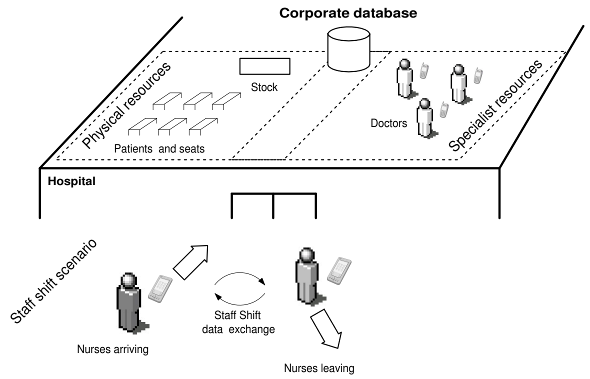

Figure 2: Mobile Service Solution to Support Team-Based Delivery of Patient-Centred Health Care

ments. This section presents a clear account of the elements and interactions in mobile service environments.

Let us consider the mobile service problem scenario depicted in Figure 2. This scenario is based on the work of [9] and can be described as follows: *a healthcare centre (hospital) seeks a mobile service solution to help to coordinate the information exchange between the support staff (nurses) during the work shifts. The requirement is to have an electronically process that keeps track of all information (formal and informal) exchanged between the nurses, thus ensuring the continuation and correctness of treatments. The organization has a* corporate database *as the central data repository. The nurses interact to this database and also exchange information with peers during the work shift. Along with information that exist in the corporate database, there is also the personal notes taken by the nurses about the patients and treatments. The solution assumes that the nurses carry a mobile computing device loaded with a mobile personal assistant application design for this task.*

It is possible to infer on the required functionality for nurse's mobile personal assistants as the application needs:

- (a) a *method to exchange data* between the mobile personal assistant between two or more nurses. For instance, the nurses need to (electronically) request the notes taken by other nurses stored in their personal devices.

- (b) a *method to sense time and location* – for instance, work shift time and near or inside the hospital – thus to infer the "context" where the nurse is and to work pro-actively for the data synchronization process. For instance, the solution must detect the presence of the nurse near the health-centre main entrance and then, pro-actively, starts the data updating procedures.

- (c)a *method to interface with nurses*, where the application presents information to the mobile user in a "user friendly" way. In the illustrative example, the personal mobile assistants running on nurses mobile computing devices must be able to interact to nurses through "intelligent" interfaces that present the *right information at the right time*, such as the list of existing patients when the nurse is arriving at the health-centre and the list of nurses arriving when near the end of their work shift.

- (d) a *method to exchange data with corporate database*, so the nurses have updated information about existing. For example, when arriving to the health-centre nurses need to have the list of existing patients and clinical paths that should be accessed by the mobile application from health-centre's corporate database automatically (and electronically).

- (e) a *method to deliberate* on application's interactions with the environment, pro-activeness and on intelligent behaviour related to interfacing with user and collaboration to other applications. For the scenarios above, the application should have a set of "deliberation rules" that control the activities in response to events of the dynamic environment (i.e., location, time) and input from nurses. These rules must reflect the activities assigned to the nurse in the organization. To provide the features of adaptivity and self-tuning the rule system must be flexible to allow the adjustment and replacement of existing rules.

The *Mobile Service Application* to be implemented for this *Mobile Service Solution* shall be equipped with the modules that implement the support to these functionalities. To properly address the above call, designers need a *reference model* through which they can answer questions such as: What are the components required to support the required functionalities? Why these components? How these components relate to the elements and interaction of the problem scenario? What are the relationship between components across multiple elements of the problem scenario? How these components should be put together to implement the *Mobile Service Application*? How the set of components in the applications for the *Mobile Service Solution*?

We propose the need to define a *framework* to guideline the development of solutions for mobile service problem scenarios. To distil this reference model, first

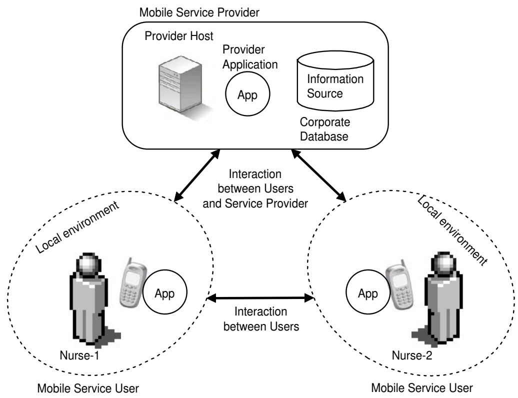

Figure 3: Conceptual Model of the Problem Scenario

we shall introduce a clear account of the elements and interactions in a mobile service environment.

Figure 3 presents the simplified view of the problem, or *Conceptual Model*, showing the active elements and interactions. It is possible to identify the *nurses* (Nurse-1, Nurse- 2) and the *corporate database* as active elements (or actors). There are interactions between nurses (to exchange personal notes about the patients) and between the nurses and the corporate database (information retrieval and update).

Next sub-section identifies the elements and interactions in the illustrative example and introduces the concept of *Mobile Service Organization*.

4 Elements and Interaction in Mobile Services

This section presents a definition of the elements and interactions in mobile service problem scenarios.

Let Org_o be a *Mobile Service Organization* named as o, which exists to deliver a mobile service to the elements of Org_o. Let $\mathcal{E}_o$ be the set of attributes of the environment of Org_o. Let $\mathcal{U}_o$ be the set of **Mobile Service Users** and $\mathcal{S}_o$ be the set of **Mobile Service Providers** in this organization.

> **(Mobile Service Organization)**
> The *Mobile Service Organization* Org is the structure:
> $Org_o = \langle \mathcal{U}_o, \mathcal{S}_o, \mathcal{I}_o, \mathcal{E}_o \rangle$,
> where $\mathcal{U}_o$ is the set of *Mobile Service Users* in the organization. $\mathcal{S}_o$ is the set of *Mobile Service Providers*. $\mathcal{I}_o$ is the set of interactions between the active elements of the organization. $\mathcal{E}_o$ is the set of attributes of the organization's environment.

In the illustrative example, the *Mobile Service Organization* for the health-Centre organization has the following elements:

- Org_{hc} is the Health-Centre organization hc.
- $\mathcal{U}_{hc}$ is the body of nurses equipped with the mobile computing device and application that enable them to access the mobile service.
- $\mathcal{S}_{hc}$ is the corporate database shared throughout the mobile service organization by an application executing on a central host, which is accessible by the mobile applications carried by the nurses.
- $\mathcal{I}_{hc}$ is the set of interactions between nurses and the corporate database.
- $\mathcal{E}_{hc}$ is the attributes of the organization's environment.

Therefore, the **Mobile Service Organization** is composed by the *active elements* (or "actors") in the mobile service problem scenario and the *interactions* between these elements.

There are two main types of *active elements* in the scenario, as described below:

- (i) the **Mobile Service Users**, which are the users, equipped with a mobile computing device loaded with a mobile service application and immersed in a local environment. In the illustrative scenario, these are the nurses carrying the mobile personal assistant.
- (ii) the **Mobile Service Provider**, which are the information providers in the mobile service solution, composed by the information source and a software application that distribute it through the mobile service environment. This is the corporate database and the infrastructure to make it accessible by the nurses in the problem scenario.

> **Mobile Service User** is a tuple
> $muser(i, u_i, d_i, a_i, E_i, I_i) \in \mathcal{U}_o$, such that: i is the identifier of the mobile user. u_i is the user. d_i is the mobile computing device carried by the user. a_i is the application running on d_i. $E_i \subset \mathcal{E}$ is the set of attributes of the local environment, which is a sub-set of the attributes of the organization's environment. $I_i \subset \mathcal{I}$ is the set of interactions between the mobile user and other elements of the organization, which is a sub-set of all interactions between the elements of the organization.

The **Mobile Service Provider** is the tuple $sprovider(j, s_j, h_j, a_j, I_j) \in \mathcal{S}_o$, such that: j is the identifier of the service provider. s_j is the provider's information source of the service provider e.g., the corporate database in the illustrative example . h_j is the host where the service's application is being executed. a_j is the service's application. $I_j \subset \mathcal{I}$ is the subset of interactions between the service provider and the elements of the organization.

Next sub-section describes the interactions between these elements.

4.1 Interactions

This sub-section describes the set of **Interactions** $\mathcal{I}_o$ of the *Mobile Service Organization* Org_o.

The diagram presented in Figure 4 provides a concise view of the elements and interaction in a *Mobile Service Organizations*. It is possible to identify **nine** relevant interactions in this environment, which are detailed below:

- (i) interaction between u_a and d_a, which are the interactions between the mobile user and the mobile computing device for a mobile service user identified as a.

- (ii) interactions between d_a and a_a, which are the interactions between device and application in a mobile service user identified as a.

- (iii) interactions between u_a and a_a, which are the interactions between user and application in a mobile service user identified as a.

- (iv) interactions between a_a and E_a, which are the interactions between application and local environment in a mobile service user identified as a.

- (v) interaction between a_a and a_b, which are the interactions between two instances of applications in mobile service users identified as a and b.

- (vi) interaction between u_a and u_b, which are the interactions between two instances of mobile service users identified as a and b.

- (vii) interaction between d_a and d_b, which are the interactions between two instances of devices belonging to mobile service users identified as a and b.

- (viii) interaction between d_a and h_b, which are the interactions between device in mobile service users identified as a and host in mobile service provider identified as b.

- (ix) interaction between a_a and a_b, which are the interactions between application in mobile service users identified as a and application in mobile service provider identified as b.

Moreover, there are also the internal interactions between the components of the *Mobile Service Provider*, which are deliberately ignored as they are not relevant in the scope of this study.

Having defined the concept of *Mobile Service Organization* and presented the definition for its elements and interactions, next section introduces the components required by intelligent mobile service applications.

5 Intelligent Mobile Service Applications

Previous section introduced the elements and interactions in mobile service environments. However, in order to build intelligent mobile service applications, the developer must integrate the components required to support the interactions between the active elements of these environments. This section introduces these components and assemble the framework for mobile service solutions.

The components of mobile service application and their relationship with the elements and interaction are detail below:

- (a) to support the interactions between application a_a and environment E_a of a mobile service user identified as a – interactions (iv) in the aforementioned list of interactions – the application requires the module for **Context Awareness**. Context-awareness enables the mobile service application to *Sense* the environment and collect relevant data, *Represent* this data as digestible information and *Infer* about the meaning of this information (e.g., if a temperature is warm or cold, or if a location is near a store). It helps in creating mobile service applications that are aware of the context where they are executing thus being able to act more proactively and requiring less user input.

- (b) to support the interactions between user u_a and device d_a, and device d_a and application a_a of a mobile service user identified as a – respectively, interactions (i) and (ii) in the aforementioned list of interactions – the application requires the module for **Device Interface**. These components enable the developer to create the application's interface that will present the information (or choices) to the mobile user.

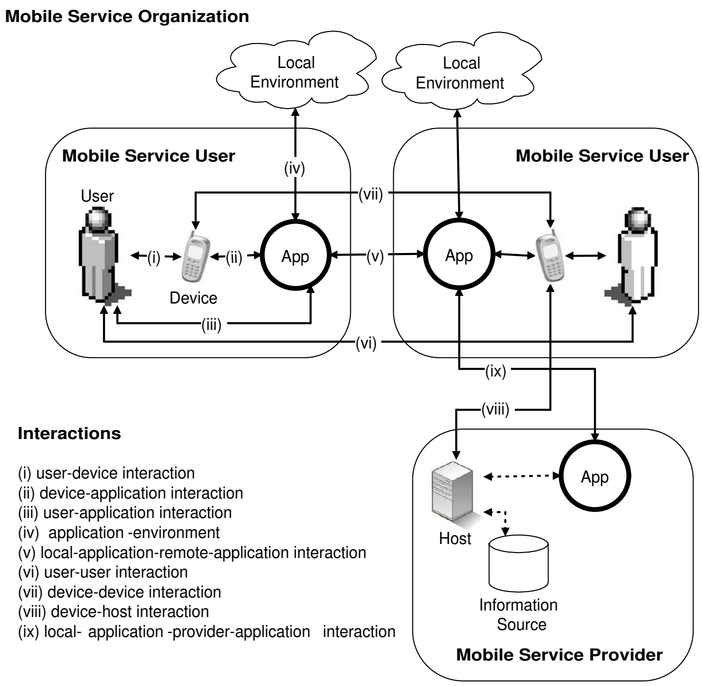

Figure 4: Elements and Interactions in Mobile Service Organization

- (c) to support the interactions between applications a_a and a_b, users u_a and u_b of mobile service users identified as a and b – respectively, interactions (v) and (vi) in the aforementioned list of interactions – the application requires the module for **Collaboration**. This module provides the solution for the issues of distribute computing, such as data communication, coordination and negotiation. The goal of this component in the proposed structure is to enable the developer to create solutions that integrate multiple users and services.

- (d) to support the interactions between u_a and a_a of a mobile service users labelled as a – interaction (iii) in the aforementioned list of interactions – the application requires a module for **Inference**. This module would also provide the support to the development and coordination of other components. The techniques to create enhanced inference systems involve the use of techniques from the field of artificial intelligence, such as the Belief-Desire-Intention paradigm. The support for enhanced inference systems has been largely explored in the field of agent-based computing, which is explore as base for implementation in this work.

Next sub-section introduces the *Generic Architecture for Mobile Service Applications*, which put the *Components* together and describe their relationships and sub-components.

5.1 Architecture

Figure 5 presents the *Framework for Mobile Service Solution*, which integrates the components presented above. This structure represents the minimum set of components required to implement intelligent mobile service solutions. The significance of this composition is to provide a complete view of the elements and modules in mobile service solutions and describe their interactions.

The *architecture* for mobile personal assistants should allow the widest possible range of mobile service solutions, creating the building blocks for intelligent mobile service solutions. The proposed approach is towards a *architecturally-neutral model*, which avoids the pitfalls of a generic architectures [10]. On the other hand, the challenge of the architecturally-neutral models is to offer enough level of detail (or to be specific enough) to provide the support for the creation of *Mobile Personal Assistants*.

Therefore, by leveraging from the guidelines provided by the *Framework for Mobile Service Applications* and towards the creation of the *generic architecture* or mobile personal assistants, as proposed by [10], it is possible to conclude that the architecture must provide the components for *information collection* (sen-

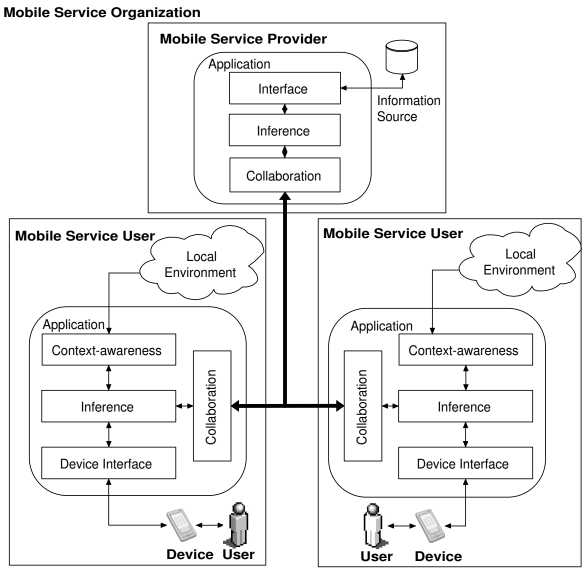

Figure 5: Framework for Mobile Service Solution

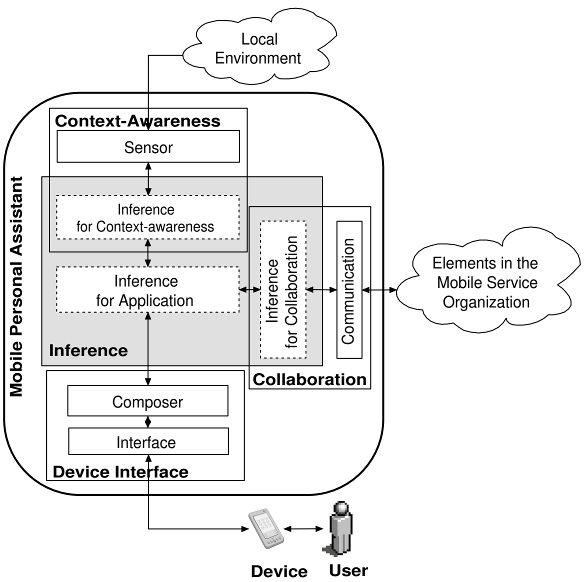

Figure 6: Architecture for Mobile Personal Assistants

sor), *information representation*, *decision-making* (controllers), *collaboration*.

Figure 6 presents the *Generic Architecture for Mobile Personal Assistants*. It provides the components to support the functionality described by the requirements aforesaid : (i) *Sensor* supports the sensor the environment (context-awareness), (ii) *Inference* supports the decision-making (inference), (iii) *Communication* supports the communication and collaboration with peer mobile users (collaboration), and (iv) *Composer* and *Interface* components support the interaction to the mobile user and environment (device interface).

The contribution from this generic architecture is towards a model that provides the guidelines for the application developer. This model provides the answer to the question about the the necessary components of the framework in order to design intelligent mobile service solutions, proposed by this work.

6 Implementing Intelligent Mobile Service Applications

The *Framework for Mobile Service Solution* works as reference model, which provides the guidelines for design decisions. The architecture for these applications must consider the support to the components presented in the framework and this reference model would inherently provide the support to the elements and interactions in intelligent mobile service scenarios. To implement these modules, the developer should use existing implementation methodologies, such as object-orientated or agent-based computing.

As mentioned, the main goal of mobile computing is to provide always-present assistance to the mobile user. *Intelligent Mobile Service Solutions* application must be able to sense the environment, process contextual information, aid in collaboration activities, sustain rational inference and equipped to provision the user with the *right information at the right time*. *Mobile Personal Assistants* are the building blocks that allow the creation of computing application equipped to deliver these features.

We argued that *Mobile Personal Assistant* can be the reference model for the implementation of agents to support the active elements in intelligent mobile service scenarios. Here, we motivate the use of agent-based software engineer to design and implement these applications.

Agents can be described as a persistent computational entity that can perceive, reason, act, and communicate. Hence, agent-based applications architecture usually present the structures as *Sensor*, *Inference, Actuator*, and *Communication* [11][12]. The parallel between the structures in agent computing and the ones required for mobile service solutions are: *Sensors* work for Context Awareness; *Actuators* work for device interface; *Communication* work to support collaboration and the inference machine, and; Belief-Desire-Intention and other techniques in agent based computing work for inference. Agents seem to offer a set of features that are closely aligned with the requirements of *Mobile Personal Assistants*. Agent concepts such as *situatedness* and *adaptivity* offer promising tools of abstraction and computational methods for building software that operates in such environments. The concept of *proactiviness* can potentially help build systems that reason about the user's goals and how they may be achieved. As pointed out by [13], agent-based software engineering provides the tools to implement these requirements, providing the solutions described below.

Furthermore, the developer must elaborate the detail design document, which describes the internals of each application component and how to accomplish its task in the overall system. The outcome from this phase are detailed documents and diagrams showing the internal functionality of each component and its capabilities, as well as descriptions of the data structures used to implement them. In order to implement agent-based *Intelligent Mobile Service Solutions* the developer must count on implementation platforms build based on the definitions proposed in the *generic architecture* mentioned early in this work.

The platform for building agents in mobile devices must provide solutions for the problems inherent to the environment, such as computing resource availability, networking, security, interfacing and compatibility. Examples of questions that arise while implementing this class of applications are: how to execute the deliberation cycle in the limited computing resources environment?; how to implement the structures for context awareness and content delivery?; what agent-oriented language to use for the development of the application knowledge structures?; which programming language to choose for the application development?

In [14] we introduce the 3APL-M (Triple-A-P-L-M) platform for implementing deliberative autonomous agents that execute on mobile computing devices. This platform is grounded on the understanding of agent systems provided through the 3APL language, presented by [15] and extended by [16]. It works as a scaled down implementation of the 3APL language interpreter re-designed for the requirements of mobile computing applications. The inference system implements the Belief-Desire-Intention paradigm. Moreover, it supplies the programming structures to deliver for the requirements of *Mobile Personal Assistants*: *sensors* for context-sensitiveness, *actuators* for content delivery, and *communication* for collaboration.

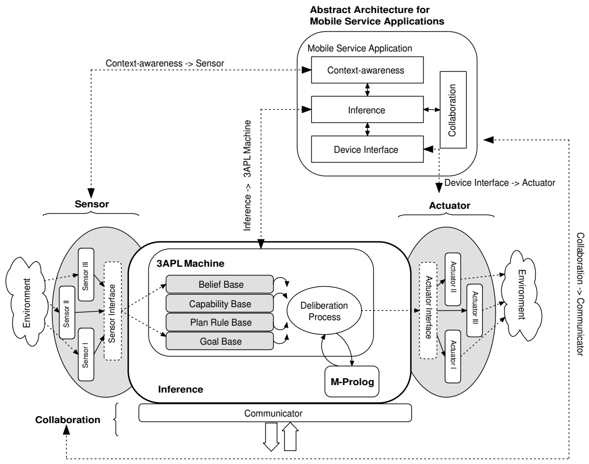

Figure 7: Relationship 3APL-M Components and Abstract Architecture Mobile Applications

Figure 7 relates the elements of the 3APL-M system architecture to the components of the *Generic Architecture for Mobile Service Applications*. The relationship between components of the 3APL-M architecture and the those of the *Generic Architecture* are as follows:

- the component for **Context Awareness** is supported by the the 3APL-M's *sensor* component along with processing rules related to capturing and representing environmental data. This component supports the interactions between mobile service application and the local environment.

- the component for **Device Interface** is supported by the the 3APL-M's *actuator* component along with processing rules related to the activities of inferring, composing and displaying information to the mobile user. This component supports the interactions between mobile user and mobile computing device and application.

- the component for **Collaboration** is supported by the the 3APL-M's *communicator* component along with processing rules related to the activities of coordination, data communication and negotiation. This component supports the interactions between applications and users.

- the component for **Inference** is implemented by the 3APL engine and supports the internal interactions in the applications.

Let us consider the development of the *Personal Mobile Assistant* for nurses in the illustrative scenario. As explained, nurse's personal assistant must coordinate the interaction between them and the corporate database. I recall that the required functionalities are to be able to: (a) exchange data between two or more nurses; (b) sense time and location; (c) interface with nurses; (d) exchange data with corporate database, and; (e) deliberate on application's interactions with the environment, pro-activeness and on intelligent behaviour related to interfacing with user and collaboration to other applications.

Figure 8 describes the solution. For the implementation of the location context-awareness, the *sensors* for *Location* could be connected to a *Location System*, based on an available technology. Once (a) the presence of a nurse near the *main entrance* point of presence is detected, (b) the application triggers a goal *sensor("location", near("main entrance"))*, whose *guard* check if the nurse is in the right context (i.e., near her work shift time). If so, the application execute the two rules related to collecting and displaying the relevant information to that point in time. Examples of the generated interfaces are presented.

In this section, we demonstrated that the definitions proposed by the introduced *Framework for Mobile Service Solutions* work as the basis for the development of implementation platforms suitable to the creation of intelligent applications.

7 Conclusion

The purpose of this work is to provide a clear definition of the modules required to design intelligent mobile service solutions. The aim is to address the lack of a framework to specify the elements of intelligent mobile service solutions. This gap leads to the use of ad-hoc solutions in expense of generality and reuse.

The paper introduces the framework describing the elements and interactions in mobile service problem scenarios. The framework contributes with a formal definition to base the architecture and development of mobile service applications. This framework provides the guidelines to develop the *Generic Architecture for Mobile Personal Assistants*, which supports the application developer with a model for the mobile application construction.

It is proposed that applications equipped with the support to context-sensitiveness, collaboration, device interface and enhanced inference would provide the structures to support the ideal of mobile computing and to create systems that deliver the right information at the right time. This structure provides the answer to the problem of supporting intelligent interactions, which leads to the creation of intelligent mobile support.

Future work shall focus on elaborate on the methods to design mobile service solutions to support multi-user scenarios. The discussion proposed the parallel between the structures in agent computing and the ones required for mobile service solutions. The use of agent-based computing for the modelling and implementation of intelligent mobile services based on the specifications presented by this work is material for future works.

Referências

[1] Fernando Koch and Iyad Rahwan. Classification of agents-based mobile assistants. In *Proceedings of the AAMAS Workshop on Agents for Ubiquitous Computing (UbiAgents)*, New York, USA, Jul 2004.

[2] Talal Rahwan, Tarek Rahwan, Iyad Rahwan, and Ronald Ashri. Agent-based support for mobile users using agentspeak(l). In Paolo Giorgini, Brian Hederson-Sellers, and Michael Winikoff, editors, *Agent Oriented Information Systems*, Lecture Notes in Artificial Intelligence. Springer Verlag, Berlin, Germany, 2004.

[3] Donald A. Norman. *The Invisible Computer: Why Good Products Can Fail, the Personal Computer Is So Complex, and Information Appliances Are the Solution*. The MIT Press, 1999.

[4] Java 2 Micro Edition (J2ME) web-site, sun corporation, http://java.sun.com/j2me.

[5] FIPA. Nomadic application support specification. Technical Report 00014, Foundation for Intelligent Physical Agents, August 2001.

[6] Norman Sadeh, Ting-Chak Chan, Linh Van, Oh-Byung Kwon, and Kazuaki Takizawa. Creating an open agent environment for context-aware m-commerce. In Burg, Dale, Finin, Nakashima, Padgham, Sierra, and Willmott, editors, *Agentcities: Challenges in Open Agent Environments*, LNAI, pages 152–158, Heidelberg, Germany, 2003. Springer Verlag.

[7] Shamimabi Paurobally, Phillip J. Turner, and Nicholas R. Jennings. Automating negotiation for m-services. *IEEE Trans. on Systems, Man and Cybernetics (Part A: Systems and Humans)*, 33(6):709–724, 2003.

[8] Hans Chalupsky, Yolanda Gil, Craig A Knoblock, Kristina Lerman, Jean Oh, David V Pynadath,

Figure 8: Case Study: Solution for Mobile Personal Assistant for *Role Nurse*

Thomas A Russ, and Milind Tambe. Electric elves: Applying agent technology to support human organizations. In Hayam Hirsh and Steve Chien, editors, *Proceedings of the 13th International Conference of Innovative Application of Artificial Intelligence (IAAI-2001)*. AAAI Press, 2001.

[9] Stefanie Kethers, Mohan Krishnamoorthy, Alistair McLean, David Sier, Anne-Marie Vercoustre, and Ross Wilkinson. Towards team-based delivery of patient-centred health care. In *Proceedings of ITiRA 2003*, Caloundra, Australia, December 2003.

[10] Ronald Ashri and Michael Luck. An agent construction model for ubiquitous computing devices. In *Proceedings of AAMAS Workshop in Agent Oriented Software Engineering*, New York, USA, 2004.

[11] Michael J. Wooldridge. Intelligent agents. *Multi-agent systems: a modern approach to distributed artificial intelligence*, pages 27–77, 1999.

[12] Stan Franklin. Autonomous agents as embodied AI. In *Cybernetics and Systems*, volume 28, pages 499–520, 1997.

[13] Fernando Koch and Iyad Rahwan. The role of agents in intelligent mobile services. In Michael Wayne Barley and Nik Kasabov, editors, *Intelligent Agents and Multi-Agent Systems: 7th Pacific Rim International Workshop on Multi-Agents, PRIMA 2004, Auckland, New Zealand, Revised Selected Papers*, volume 3371 of *Lecture Notes in Computer Science*. Springer-Verlag GmbH, March 2005.

[14] Fernando Koch, John-Jules Charles Meyer, Frank Dignum, and Iyad Rahwan. Programming deliberative agents for mobile services: the 3apl-m platform. In *Proceedings of AAMAS'05 Workshop on Programming Multi-Agent Systems (ProMAS'2005).*, 2005.

[15] Koen V. Hindriks, Frank S. De Boer, Wiebe Van Der Hoek, and John-Jules Ch. Meyer. Agent programming in 3APL. *Autonomous Agents and Multi-Agent Systems*, 2(4):357–401, 1999.

[16] Mehdi Dastani, Frank de Boer, Frank Dignum, and John-Jules Meyer. Programming agent deliberation: An approach illustrated using the 3APL language. In *Autonomous Agents and Multi-Agent Systems 2003 (AAMAS'03)*, Melbourne, Australia, 2003.