International Journal of Software Engineering and Its Applications
Vol. 9, No. 11 (2015), pp. 227-236
http://dx.doi.org/10.14257/ijseia.2015.9.11.21

Design of Mobile Collaboration Frameworks

Haeng-Kon Kim

Dept. of Computer Information & Communication Enginerring,
Catholic University of Deagu, Korea
hangkon@cu.ac.kr

Abstract

The recent years have shown a tendency where computers have become more and more portable to fit the needs of users on the move. This tendency has further been strengthened by the growth in the mobile phones user base in parallel with the increasing computational capacity of the mobile phones. Today, mobile phones have the same computational capacity of the late 80's computers with one important addition. Mobile phones are constantly connected to a network, making them available for collaborative work anywhere and anytime.

In this paper, we suggested a mobile collaboration framework based on distributed object group framework. This paper will aim to design and implement a framework used to create collaborative applications on mobile phones. The emergence of mobile phones with some kind of ad hoc network technology built-in creates opportunities for new forms of computerized collaboration. Mobile, computer supported, collaboration between people that are collocated is now possible.

Keywords: *u-health Agent node, Mobile collaboration frameworks, Agent UML modeling, Mobile development*

1. Introduction

Research on computer supported cooperative work recognized from the very beginning two main cooperation modalities - face-to-face and remote cooperation - and tried to understand their nature and to support them through effective applications. More recently, the concept of local mobility has been introduced as an in-between modality between face-to-face and remote cooperation, and as something that may cause the interruption of cooperative activities [1], but also as a property of most work situations [2]. Of course, this effects another fundamental aspect of cooperation, that is *continuity* both in the physical space (no matter if a person stays at her desktop or gets up and walks to another room, her work must be supported without discontinuity) and in the logical context of action (for example, if a person is working on a document with a coworker, they don't have to abandon their activity to find information related to the document). In our view, the shift from the desktop computer metaphor to the mobile computing one is promising in the aim to support cooperative work with a smooth form of coordination; in fact, people become able to act and interact, in a more natural and instinctive manner, within a computational environment that is aware of persons and activities and that is able to adapt the support it provides to the changing context. To reach this goal the integration of the themes of cooperation and the mobile computing paradigm has to be strengthened, more than it is currently. In this view, we propose the notion of *community*, so the notion of community varies from "interest groups" up to "strategic communities" as a first class concept of this integration for its suitability in representing the plasticity of cooperation. In fact a community is spontaneously built and legitimates various degrees of participation of (new) people on the basis of its internal rules: this is usually called

ISSN: 1738-9984 IJSEIA

"legitimate peripheral participation". Beside this, a community autonomously organizes and builds its memory and interactions with other communities to enhance its capabilities as well as with the institutional context in which it operates.

The design of mobile-computing environments to support cooperation requires a reference model able to take into account the above specifications and in particular the notations of community and flexible peripheral participation. To our knowledge such model has not yet been proposed, since the cooperation dimension is usually totally disregarded or left implicit and community is not a first class object. The aim of this paper is to give a contribution in this direction by defining an approach to conceive cooperative work that is inspired by the mobile computing paradigm and by proposing a new model as a basic step toward architecture able to adequately support the design of collaborative ubiquitous-computing environments [3, 4]. Development of applications requires a lot of effort not only from many stakeholders at implementation time, but also when it comes to capturing domain concepts, understanding underlying technology, and creating the necessary infrastructure for the application. Since the market for development of mobile applications is quite young and the technology itself is quite immature, there are not many common building blocks available that can be used to build the actual applications. Frequently this leads to developers reinventing the wheel by implementing almost similar infrastructures. Many research projects in the field of mobile collaboration have resulted in specialized prototypes demonstrating the possibilities of the available technologies. No one has yet made a general framework for these types of applications on mobile phones. The researchers spend a vast amount of time developing these prototypes because everyone is starting from scratch, developing their own unique architectures.

The design of mobile-computing environments to support cooperation requires a reference model able to take into account the above specifications and in particular the notions of community and flexible peripheral participation. To our knowledge such model has not yet been proposed, since the cooperation dimension is usually totally disregarded or left implicit and community is not a first class object. The aim of this paper is to give a contribution in this direction by defining an approach to conceive cooperative work that is inspired by the mobile computing paradigm and by proposing a new model as a basic step toward architecture able to adequately support the design of collaborative ubiquitous-computing environments.

In this paper, we suggested a mobile collaboration framework based on distributed object group framework. This paper will aim to design and implement a framework used to create collaborative applications on mobile phones. The emergence of mobile phones with some kind of ad hoc network technology built-in creates opportunities for new forms of computerized collaboration. Mobile, computer supported, collaboration between people that are collocated is now possible.

2. Related Works

2.1. MC (Mobile Computing and Cooperation)

Mobile computing (MC) is still more an idea than a reality since embedding computation into real everyday objects is not a simple task from the technical point of view and it is usually achieved either in prototypes or quite expensive devices. However, the rapidity and unpredictability of the technological evolution suggest playing with this idea to be ready when it becomes feasible and be able to master the implications in application design. So, in the following we will consider some of the implications of MC without been too much constrained by the current technological achievements. We suppose that each object can have specialized computational capabilities making them reactive and proactive in relation to actors and/or other objects that are close of in their surroundings. This distributed computing power is connected through a wireless network that supports bi-directional information flows towards and from more traditional computational nodes (usually standard PCs).

Which is the role of MC in supporting cooperative work, both at the individual and collaborative dimension? Both of them are relevant since both have to be smoothly integrated in technologies that support cooperation. The MC literature is generally more focused on the individual dimension, and only recently it contained an explicit suggestion to consider cooperation in MC environments [5, 6]. A way to connect MC and cooperation is through the notion of context, since UC and context-aware computing share the same goal to make the environment "alive" and its context an important part of what determines the application's behavior. More specifically, we like the idea to view context not as a representational problem but as an interactional problem, as proposed in: context has to be seen as a relational property that holds between objects or activities; something may or may not be contextually relevant in relation to some particular activity; the scope of contextual features is defined dynamically; context is an occasioned property, relevant to particular settings, particular instances of action, and particular parties to that action; finally, context is not out there rather it arises from the activity.

Figure 1 shows the system architecture to support the mobile collaborations between different devise users as PC, laptop and mobile phone, We focus on the mobile and mobile users collaborations in this paper.

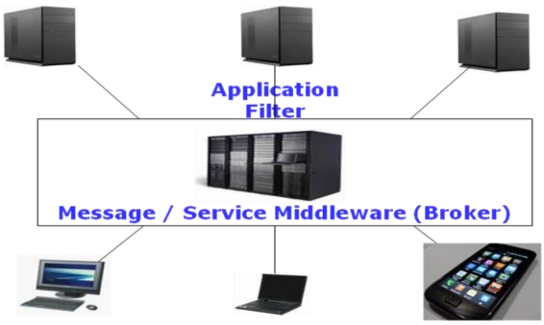

Figure 1. Collaboration Computing Environment Architecture

2.2. Mobile Device User's Collaborations

Developing applications for mobile users is difficult and problematic. A simple comparison between the resources available to the average desktop user and their mobile counterparts starkly illustrates this. Processing power, screen size, battery or power source life span and network capability are all significantly poorer on PDAs when contrasted with their desktop cousins. Indeed, it is probable that this resource gap will always exist. Likewise, methods of interacting with portable devices differ from those traditionally used. Unfortunately, the issue of solution deployment is further complicated when it is considered that portable devices can differ significantly between themselves. Despite such limitations, it is the dynamics of both the mobile user's behavior and the environment within which the system is being used which form the more interesting challenges and, as such, prove of particular interest to us. A mobile user's context is by definition dynamic as their needs and their environment are in a continual state of flux. This implies that their needs and expectations may change as they roam about the physical environment. In addition, the quality of what we might term the prevailing electronic infrastructure, for example networking ability, may vary considerably at any given time. How best to engineer applications that can operate under such varying circumstances while simultaneously meeting users' needs and expectations is still an open question [7].

3. Modeling of Mobile Collaboration Framework

This section presents an overview of our MC (Mobile Collaboration) framework architecture. This framework provides flexible support for defining and configuring the group through the service components of mobile and behavior of the principal elements of a mobile collaboration environment. Also, we describe the type of mobile collaboration. An MC collaborative environment is composed of objects (we also include devices in this term) that show a variety of computational capabilities: from sensors to wall-boards, from documents to pieces of furniture, from desks to doors, and so on, up to traditional general purpose computers that can play the hidden role of servers or the visible role of terminals supporting individual work.

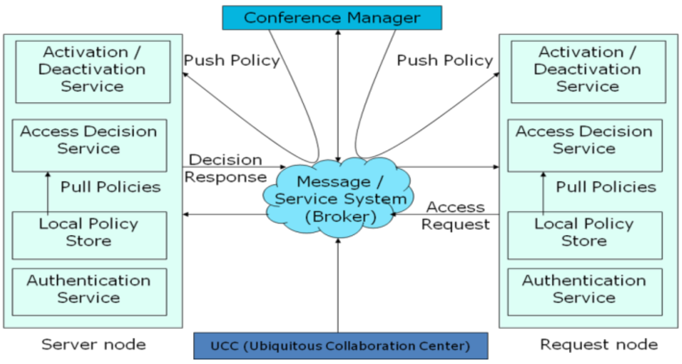

Figure 2. Frameworks for Mobile Collaboration

A part from MCs, each object is dedicated to a specific functionality that can be provided by local computational power or by the interaction with a computational environment offering (mobile) connectivity. As alluded earlier, it is difficult to foresee the future technological development: however, the second case is more likely as long as the environment's reactive and proactive behaviors become more challenging. The smart behavior of these objects requires the availability of computational capabilities that are hidden and fully independent from the actors moving in the smart environment. The latter has to guarantee a service to objects acting as sensors and actuators, which is characterized by quality levels that can be achieved by purely architectural considerations: computation efficiency, reliability, robustness, and so on. On the other hand, the smart behavior of the individual objects and of the environment as a whole requires the availability of distributed inference capabilities able to transform elemental data into more complex information about people and the physical setting, up to the construction of knowledge on which to base smartness.

3.1. MC Architecture

The collaborative dimension adds further requirements to recognize and support the communities acting in the MC environment. In fact, people freely move in the physical space carrying and approaching objects that altogether provide different forms of computational connectivity, as well as meet other people and establish with them various forms of cooperation. Moreover, considering the logical dimension, which is the dimension where information and coordination resources are managed to support these forms of cooperation, we can recognize a similar kind of dynamism: actors own, make available, approach and coordinate their access to these resources in a flexible way according to their needs, interests, duties or simply because they realize that an opportunity is offered to them or that the current state of affairs requires their coordinated intervention. The degree of participation of a person to a community is proportional to the person distance from the center of the community, *i.e.*, from the locus where the (physical and/or logical) ties linking its members are stronger. The above mentioned distributed inference capability should also recognize and support modulated participation of actors in the different kinds of communities by enforcing the rules characterizing them and facilitating coordination without communication. Figure 3 shows the general work context in mobile collaboration [13]. In our work, we also consider it as layered architecture. Each specific work context provides requirements that should be addressed by computing devices and the software application to be used. Particularly the environmental factors (*e.g.*, uncomfortable workspace) and the features of the users' activity (*e.g.*, massive data input) provide *technological requirements* to be satisfied by the computing devices to be used. On the other hand, the features of the users' activity and the users' goals (*e.g.*, fast reviewing of a whole document), which are part of the same specific work context, provides *functional requirements* to the software application supporting the activity. The proposed evaluation framework acts as an instrument allowing mobile groupware applications developers to match technological requirements with functional ones in order to determine: (a) advantages and disadvantages of every type of mobile computing device to support the application functionality in such work context, (b) which variants of a software application need to be developed in order to cover the specific work contexts, and (c) what functionality could be included in each variant.

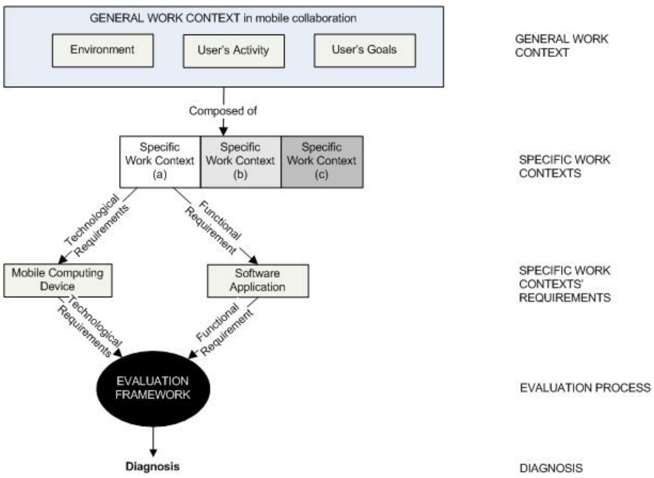

Figure 3. General Work Context in Mobile Collaboration

The characteristics of agents, though the subject of much debate in the past, have coalesced around a number of core features. These include autonomy, reactivity, proactivity and co-ordination. If we consider a typical mobile user for example, an agent might operate on their PDA, monitor the user_s behavior in an autonomous fashion, react to any perceived changes in the user_s status, and proactively anticipate what the user_s future behaviour will be. Finally, in co-ordination with other agents, it might arrange access to those services that it thinks the user may need as their context changes. While such characteristics satisfy the criteria of agenthood for many people, some researchers consider them as being only the minimum required. Indeed, such people would augment these characteristics with other stronger features such as sophisticated reasoning capabilities. For the purposes of this article we subscribe to this latter view and utilise agents that have been realised through the BDI paradigm [3]. BDI agents are a classification of agents constructed around the concepts of Beliefs, Desires and Intentions. To summarise briefly: an agent is assigned a number of tasks to perform. Such tasks essentially form the agent_s raison d_eˆtre and, in the BDI scheme, are represented as desires. However, before a desire can be fulfilled, a number of criteria must be satisfied. Such criteria, referred to as beliefs, are continuously formulated and updated as the agent monitors its environment. In general, it is unlikely that an agent will be able to fulfill all its desires at any given time. However, it is realistic to assume that it could fulfill some of them and such desires are formulated as intentions which the agent then proceeds to undertake. BDI agents are a mature realisation of the strong intentional agent stance [4] and offer an intuitive and computationally tractable model of intelligent agents. Naturally, this makes using the paradigm attractive to those designing and developing agent-based solutions. Our framework used to component of supporting object group management for domain grouping in distributed object group framework, we consider the interaction of mobile devices and sensors.

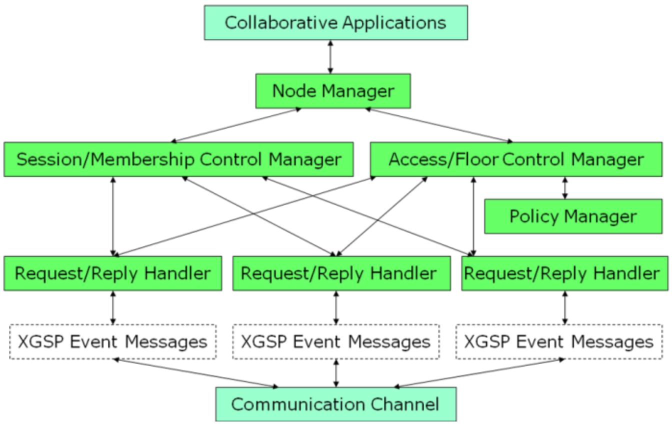

Figure 4. Collaboration Applications Layer in this Paper

Also, for information collection and sharing in this environment, we adopted the TMO scheme and TMOSM [12] into the development environment of the healthcare application. Figure 5 shows its architecture. It supports a logical single view system environment by grouping them. The group manager API supports the execution of application of appropriate mobile collaboration service on upper layer by using the input information obtained from individual or grouped physical devices on the lower layer as distributed platform. That is, according to the mobile collaboration service or status of the logical domain for collaboration, our framework could configure new groups dynamically by integrating physical devices/sensors or machines on the distributed platform and distributed application and agent on the upper layer.

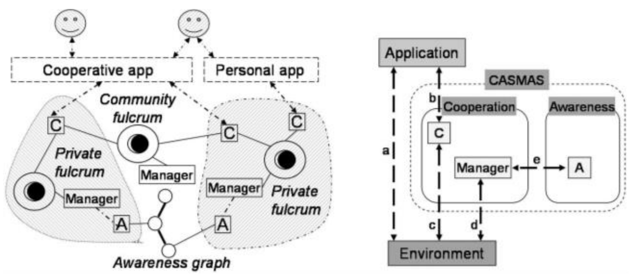

Figure 5. Architecture of MC Framework in this Work

3.2. The interactions of Components

In the whole mobile collaboration framework, we defined the interaction of components which interacts with the distributed application include the agent, sensors and components of framework. The group manager object provides the interaction of agent in distributed application on mobile devices or Server and sensor by APIs and service object reference which support collecting real time information from the sensor nodes. Also it supports the security service which is a security object that checks access right for client. When service object is replicated, dynamic binder object provides the reference of service object by binding algorithm. The stationary-typed agent on mobile devices obtains real time information of sensor node through service object reference. And, the interaction of objects in distributed application returned the result of service by framework components. The suggested framework located at home server supports the security service and manages the devices and related information. Also, home server supports management of collecting information by stationary-typed agent and control of appliance by the context provider object. Figure 2 shows the sequence diagram of relationship with distributed applications, agents and component objects for framework.

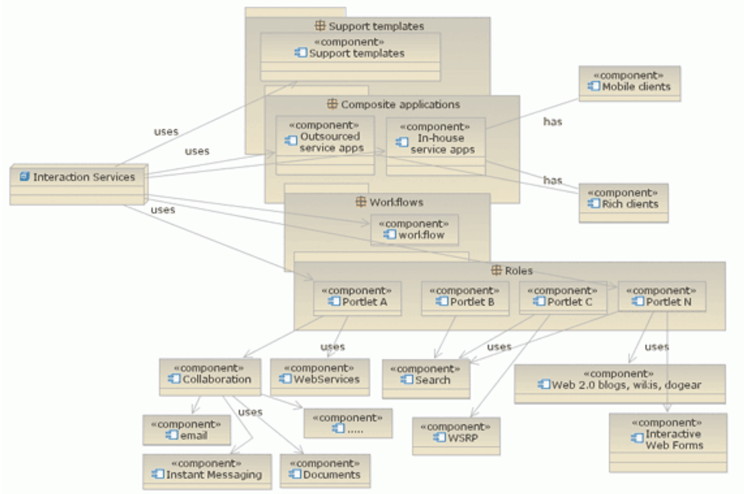

Figure 2. Class Diagram for Mobile Collaboration

3.3. The Agent Type of Mobile Collaboration

This paper suggested mobile collaboration framework, interacts with devices and sensors according to the type of devices. According to the function, we divide it into two classes; stationary and moving-typed agent. The functions of a stationary-typed agent are collecting related patient health data and environment information about the ward from the healthcare sensors or devices. The moving-typed agent can support the nurse business. Also, its function is similar to the stationary-typed agent. The difference between stationary-typed agent and moving type agent is that the latter only collects the data whereas the moving type agent usually provides the interaction with other agents and home server.

We defined the mobile collaboration type about the interaction of devices. The information collected by sensors can be shared and exchanged by agents or home server in accordance to Push and Pull methods. The interaction between sensors and stationary-typed agent is using push methods that is collecting information using sensor send to moving-typed agent. And, the interaction between stationary-typed agent and moving-typed agent is using push/pull methods. In this case, the push methods are available to reconfiguration for sensor network and collecting the data from the sensor nodes. Then the pull methods are available to the exchange information between agents. Also, the mobile device and home server used push/pull methods of agents and communication ways which can use Bluetooth or wireless LAN. Firstly, Bluetooth way is to exchange data with the home server which is searched Bluetooth dongle. When mobile device has no Bluetooth dongle for the communication way, then mobile device can search the access points by IEEE 802.11b, g. After searching for several devices by AP, it is communicating with searched devices. The above interaction way is according to the system environment. The home server provides information management and service based on context recognition. Figure 6 shows the architectures for the agent type id mobile collaboration in this paper.

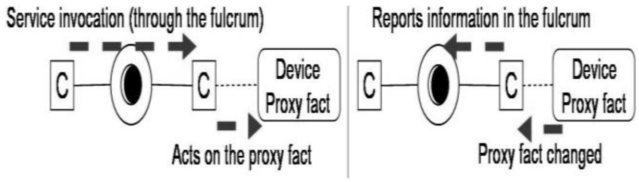

Figure 6. Architectures for the Agent Type of Mobile Collaboration

3.4. Designing of Healthcare Applications using the Framework

It is possible to identify the best mobile computing device to support collaborative activity in context by considering the requirements from each specific work context. The first step is to identify which of the features. The components of the healthcare application based on mobile collaboration framework are defined by the TMO scheme. And we used the TMOSM for interactions between distributed components. Figure 4 describes the interaction of the components for healthcare application.

For the healthcare application, distributed application components of the system(Client, Server) includes the distributed object implemented by TMO scheme and agents that autonomously act according to the perceived context information.

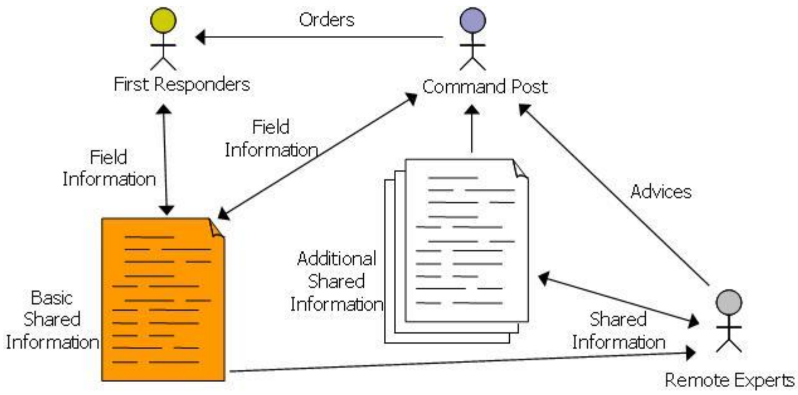

Figure 4. The System Environments of Application for Mobile Collaboration

In order to facilitate the implementation of autonomous agent for this healthcare Service, we used one more TMOs. Sensor TMO collects environment information, patient health information at the wards and also obtains the location information by agent. Monitoring GUI display the information about the collected data from Sensor TMO. Profile TMO manages user profile, user authority information and Control TMO have responsibility for appliance control. The Context TMO in home server, component for the appliance control service, monitors the action of all information appliances by receiving the information from corresponding appliances. The stationary-typed agents are located on MOBILE DEVICES s at wards and home server which are fixed host. The moving-typed agent is located on mobile devices which is mediate component that interact with mobile devices and PCs. First, for the moving-typed agent on mobile devices, the Sensor TMO mapping to moving object, called a nurse, in hospital, and sensed by physical sensor (Cricket). It also senses the moving nurse by the periodic time description, stores the location information of home server into information repository.

When detecting the moving object, Sensor TMO transfers the location information. And it obtains environment information and patient's health information from sensors or stationary-typed agent at ward. Control TMO controls peripheral appliances by sending those commands in their own proprietary language. In this case, the security object in DOGF needs to grant the access control right this privilege. All of the access control rights specified for the security object need this access privilege.

For stationary-typed agents on mobile devices at wards, they also collect data about the environment information and patient's health information from the environment sensors and health sensors. And then, the information transfers to the Sensor TMO in moving-typed agent.

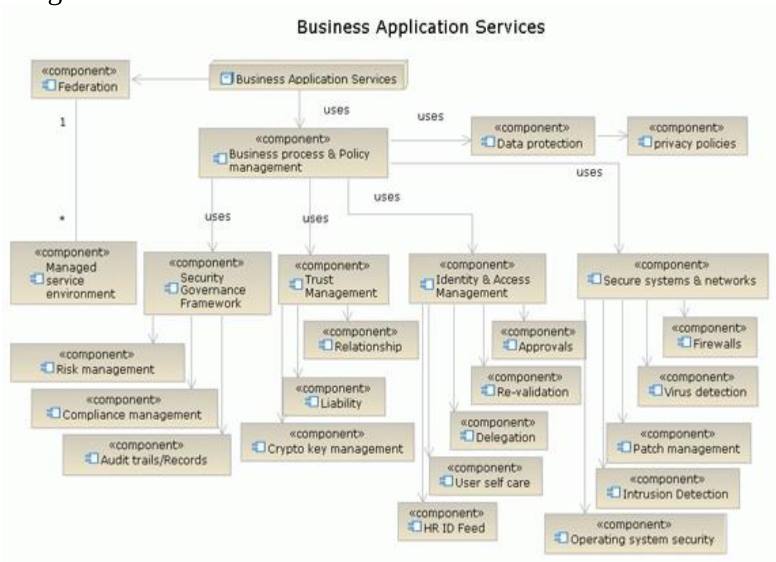

Figure 5. Business Application Service Class Diagram in Our Works

4. Conclusion and Future Works

Mobile computing environment includes the various sensors, mobile devices and communication infrastructure. In this environment, research is towards the mobile collaboration that able to deal with the mobility and interactions of both users and devices. Hence, because of limitations on resource and platform, we suggest mobile collaboration framework. It is based on the distributed object group framework that supports the group service and real time service. The mobile collaboration environment includes the sensors, mobile devices and home server. We defined the interaction agent type that interacts with each other. And we applied healthcare service for mobile collaboration such as a hospital ward environment. Each component for executing functions of agent and an integrated monitoring is implemented by TMO scheme. And we used the TMOSM for interactions between distributed components. Finally, we showed via GUI the executablility of healthcare application on our mobile collaboration framework. Our future work will apply different environments for mobile healthcare service and improving the performance of the framework. We will include studying the mobile agent technologies and then, we will apply to the moving-typed agent.

Acknowledgments

This research was Supported by the MSIP (Ministry of Science, ICT and Future Planning), Korea, under the C-ITRC (Convergence Information Technology Research Center) support program (IITP-2015-H8601-15-1007 (1711026590)) supervised by the IITP (Institute for Information & communication Technology Promotion).

This research was also supported by the International Research & Development Program of the National Research Foundation of Korea (NRF) funded by the Ministry of Science, ICT & Future Planning (Grant number: K 2014075112).

References

[1] C.-S. Shin, C.-W. Jeong and S.-C. Joo, “Construction of Distributed Object Group Framework and Its Execution Analysis Using Distributed Application Simulation Embedded and MobileComputing”, International Conference EUC, **(2004)**, pp.724-733.

[2] D.-S. Kim, C.-W. Jeong and S.-C. Joo, “Implementation of Healthcare Application Service in Mobile Collaboration Environment”, Proceedings of Korea Computer Congress, vol. 33, no. 1D, **(2006)**, pp. 88-90.

[3] A. Heinemann, J. Kangasharju, F. Lyardet and M. Mühlhäuser, “iClouds - Peer-to-Peer Information Sharing in Mobile Environments”, International Conference on Parallel and Distributed Computing (Euro-Par 2003). Klagenfurt, Austria, **(2003)** August 26-29.

[4] “Sun Microsystems”, jxme : JXTA Java Micro Edition Project, http://jxme.jxta.org/.

[5] R. W. Collier, *et al.*, “Beyond prototyping in the factory of agents”, in: Proceedings of the 3rd International Central and Eastern European Conference on Multi-Agent Systems (CEEMAS_03). Prague, Czech Republic, Lecture Notes in Computer Science (LNCS), Springer Verlag Publishers, **(2003)** June.

[6] “Agent UML”, Available, http://www.auml.org.

[7] F. Bergenti and A. Poggi, “LEAP: a FIPA platform for handheld and mobile devices”, in: Proceedings of the 8th International Workshop on Agent Theories, Architectures and Languages (ATAL), Seattle, WA. USA, **(2001)** August.

[8] “Foundation for Intelligent Physical Agents (FIPA)”, http://www.fipa.org.

[9] D. Johansen, *et al.*, “A tacoma retrospective”, Software Practice and Experience, vol. 32, no. 6, **(2002)**, pp. 605-619.

[10] “The Agent Oriented Software Group”, http://www.agent-software.com.

[11] G. M. P. O_Hare, M. J. O_Grady and G. Genie, “A multi-agent system for mobileand intelligent content delivery”, Computer Communications, vol. 26, no. 11, **(2003)**, pp. 1177-1187.

[12] A. Neyem, S. F. Ochoa, J. A. Pino and R. D. Franco, “A reusable structural design for mobile collaborative applications”, Journal of Systems and Software, In press, **(2011)**.

[13] L. A. Guerrero, S. F. Ochoa, J. A. Pino and C. A. Collazos, “Selecting Computing Devices to Support Mobile Collaboration”, Group decision and negotiation, vol. 15, no. 3, **(2006)**, pp. 243~271.