International Journal of Engineering Research & Technology (IJERT)
ISSN: 2278-0181
Vol. 4 Issue 01,January-2015

# Multi-Agent System for Work Orders Management Based on Android Operating System

Rachida Abidar, Kamal Moummadi
Team of Systems' Architecture
Laboratory of Computing, Systems and Renewable Energy,ENSEM
Casablanca, Morocco

Hicham Medromi
Team of Systems' Architecture
Laboratory of Computing, Systems and Renewable Energy, ENSEM
Casablanca, Morocco

***Abstract*—The recent developments of mobile communication systems, location service, new intelligent mobile devices, structuring and data access, have made possible the development of new services and applications making daily life easier by generalized access to real time information. This paper presents an intelligent data synchronization platform able to manage efficiently the work orders of the Company of Management and distribution of water and electricity. The platform ensures the work orders (WO) exchange in real time between the company Information System (IS) and PDA of field operators. Synchronization platform takes into account the location of the field operators and assigns work orders by using the GPS location service available in the PDA. The proposed platform is formed by two components: central software component (CSC) integrated into the IS and Embedded Software Component (ESC) on the PDA. This platform is based on multi-agents approach, mobile agent adapts well with constraints imposed by the mobile device with limited resources such as PDA. JADE-LEAP android is used to develop the ESC and the communication between these two components is performed using the Secure Socket Layer (SSL) protocol implemented using the JADE-S plug-in.**

**Keywords—*Multi-Agent Systems; JADE-LEAP android; mobile device; work orders management; real time synchronization; GPS.***

## I. INTRODUCTION

The development of communication systems and location service and progress of mobile devices make possible the implementation of new applications and services, allowing access to information in real time.

In this paper we present an intelligent data synchronization platform able to manage efficiently the work orders of a company of management and distribution of water and electricity. Indeed, the existing solution does not provide real-time data synchronization between the information system and PDA. At night, the operators move to company, to store work orders in the IS and to load the WO for the next day on their PDAs, through a wireless network. This system is not flexible: scheduled work for the operator cannot be changed, because an urgent intervention is requiring.

The proposed architecture allows improving business processes by ensuring the capability of real-time WO delivery to mobile devices. Synchronization platform takes into account the location of the field operators and assigns WO by using the GPS location service available in the PDA. This will allow optimizing and minimizing intervention time, and reduce unnecessary move to and from company, and increasing the field operator efficiency.

The platform ensures the WO exchange in real time between the IS company and PDA. The data is in text form. Indeed, data synchronization often involves frequent processes which must tend toward real time. It is therefore imperative to reduce data volumes exchanged to reduce the processing time.

The proposed platform is formed by two components: central software component (CSC) integrated into the IS and Embedded Software Component (ESC) on the PDA. This platform is based on multi-agents approach, mobile agent adapts well with constraints imposed by the mobile device with limited resources such as PDA. JADE-LEAP android is used to develop the ESC and the communication between these two components is performed using the Secure Socket Layer (SSL) protocol implemented using the JADE-S plug-in.

The rest of the paper is organized as follows: Section II starts with the description of intelligent agents, Multi agent system (MAS) and an overview of jade-LEAP platform. The proposed architecture will be described in Section III. Section IV presents the intelligent platform of WO management used to validate our architecture. The final section concludes the paper and takes a look at future work.

## II. MULTI AGENT APPROACH

### A. *Intelligent agent*

Intelligent agents can be viewed as autonomous software or hardware constructs that are proactively involved in achieving a predetermined task and at the same time reacting to its environment. According to [1], agents are capable of:

- Autonomy, they can perform their tasks without a direct and continuous guidance from the user.

- Learning, they can apply machine learning techniques to construct automatically a user profile and adapt their actions to the user's preferences.
- Proactivity, Agents can be proactive, in terms of being able to exhibit goal-directed behavior, reactive; being able to respond to changes of the environment, including detecting and communicating to other agents, autonomous; making decisions and controlling their actions independent of others [2].
- Collaborative behavior, n agent can work in concert with other agents to achieve a common goal [2],thus; an agent is part of a multi-agent system. The agent must be able to communicate with the user or the device that it represents and with other agents in order to collectively tackle problems that no single agent can handle, individually.
- Mobility, an agent is able to migrate in a self-directed way from one host platform to another [2].We identified this characteristic on agents that require changing or updating their reasoning behavior when certain conditions in the environment change.

*B. Multi agent systems*

Multi agent systems are a new paradigm for understanding and building distributed systems, where it is assumed that the computational components are autonomous.

The use of agent systems to simulate real-world domains may provide answers to complex physical or social problems that would otherwise be unobtainable due to the complexity involved, as in the modeling of the impact of climate change on biological populations, or modeling the impact of public policy options on social or economic behavior. In addition, multi-agent models can be used to simulate the behavior of complex computer systems, including multi-agent computer systems. Such simulation models can assist designers and developers of complex computational systems and provide guidance to software engineers responsible for the operational control of these systems. Multi-agent simulation models effectively provide a new set of tools to manage complex adaptive systems, such as large-scale online resource allocation environments [14].

We do not claim that agent systems are simply panaceas for these large problems; rather they have been demonstrated to provide concrete competitive advantages such as:

- Improving operational robustness with intelligent failure recovery;
- Reducing sourcing costs by computing the most beneficial acquisition policies in online markets; and improving efficiency of manufacturing processes in dynamic environments.

The MAS possesses the traditional advantages of the distributed and rival resolution of problems [3] [19]:

- The modularity allows returning the simpler programming. She allows, furthermore, the MAS to be easily stretchable, because it is easier to add new agents to MAS than to add new capacities to a monolithic system.
- The speed is mainly due to the parallelism, because several agents can work at the same time for the resolution of a problem.
- The reliability can be also reached, as far as the control and the responsibilities being shared between the various agents, the system can tolerate the failure of one or several agents.

*C. Agents on mobile device:*

A number of efforts have been made at deploying intelligent agents on mobile devices including PDAs and smart phones. There are several multi-agent platforms for mobile devices such as MobiAgent [5], AgentLight [6], MicroFIPA-OS, Agent Platform [7], and jade-LEAP [4]. We choose the jade-LEAP platform for many reasons such as: [8]

- Extension to JADE which written in pur java, and have features such as the possibility of executing multiple concurrent tasks (behaviors) in a single Java thread, matched well the constraints imposed by devices with limited resources. [4]
- Supports large variety of devices, it can be deployed not only on PCs and servers, but also on lightweight resource devices such as Java enabled mobile phones. Smallest available platform in terms of footprint size,
- Proprietary device-initiated and socket based communication channel with main container,
- Developed within LEAP project,
- Open-source,
- Supports the deployment of agents on a heterogeneous network of fixed and mobile devices.

*D. Overview of JADE-LEAP ANDROID*

JADE-LEAP ANDROID is a JADE add-on that provides support for using JADELEAP on Android Platform. Android is the software stack for mobile devices including the operating system released by Google within the Open Handset Alliance. The possibility of combining the expressiveness of FIPA communication supported by JADE agents with the power of the ANDROID platform brings a strong value in the development of innovative applications based on social models and peer-to-peer paradigms. The JADE-ANDROID add-on has been designed to support this kind of applications. By means of it, an ANDROID application can easily embed a JADE agent and therefore become part of a wider distributed system possibly including other mobile devices (not necessarily ANDROID enabled). More in details the add-on provides an interface that allows the application to start a local agent, trigger behaviors and more in general exchange objects with it.

It is therefore possible to discover remote peers, carry out probably complex conversations with them, and exploit the JADE ontology support to handle structured messages, perform background activities according to the behavior composition model and, in general, take advantage of all features of the JADE platform.

In order to be compatible with the Dalvik JVM and to properly cope with the limitations and constraints of mobile devices and wireless networks, the add-on makes use of the JADE-LEAP version for Personal Java and the split-container execution mode. This allows limiting communications over the wireless link as much as possible and obtaining very fast communication between mobile peers.

## III. PROPOSED ARCHITECTURE

As shown in Fig. 1, the proposed architecture is composed of 4 layers. Each layer consists of one or more multi-agent systems. We proposed a modular architecture and we consider the security aspect.

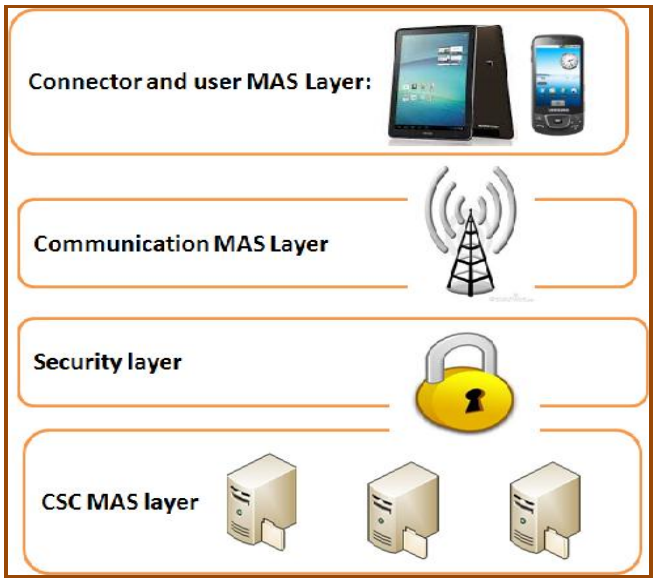


Fig. 1. Abstract view of proposed architecture

Fig. 2, illustrates differents MAS of the proposed architecture, these MASs implement two main components, ESC and CSC [20].

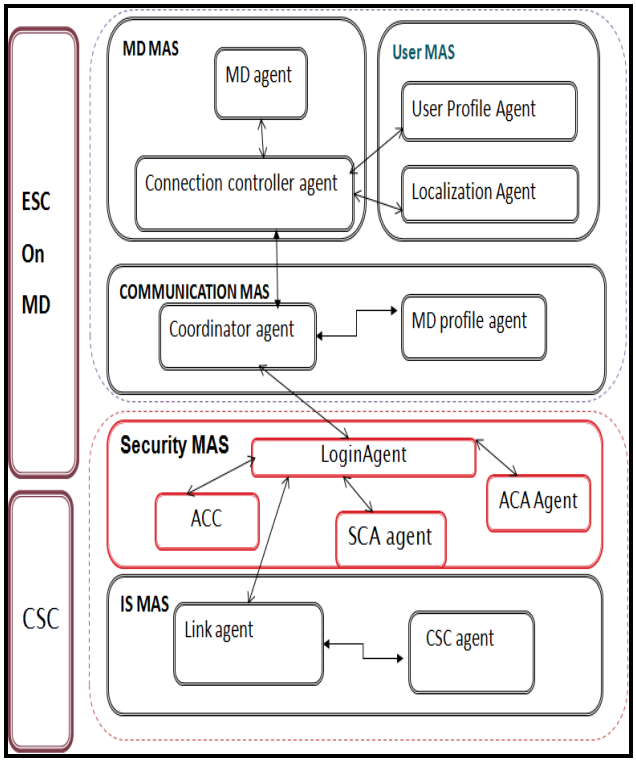


Fig. 2. proposed architecture

### A. *Embedded software component (ESC):*

It is an application embedded in the form of a service which turns permanently tuned to new data to synchronize them with CSC. The synchronization is realized in an automatic way without intervention of the user: once the PDA is switched on, the synchronization is automatically established, and if the ESC receives a new data, a notification appears in the taskbar. ESC is composed of several agents [13] [14] [15]:

II. *MD MAS:* a collaboration of several MD agents and a connection controller agent. MD Agent is running on the user's mobile phone and it provides a graphical interface that allows the user to send requests and to see the result of his request. Connection controller agent: detects different connection types over MD, and chooses the appropriate one depending on bandwidth and debit …and then sends the user request.

*III. Communication MAS:* provides an interface that makes transparent communication between users and IS. It is composed of a MD profile agent and a coordinator agent.

- *MD profile agent* groups the characteristics of MD, among which there are various aspects such as technical constraints and connection of MD, and its behavior in certain types of network, etc…
- *Coordinator agent* is in constant communication with the connection controller agent to check the connection status of DM agents. If there are problems with the connection controller agent, for example, if the connection controller agent fails, the coordinator agent will attempt to temporarily store data in a local database SQLite on the MD.

*1) User MAS:* provides functionalities related to users. It is composed of a user profile agent and a localization agent.

- *User Profile Agent*: manages information about the user characteristics (identity, role, etc.)
- *Localization Agent*: captures the location of the user through the GPS, to adapt the query result to the user's location. GPS is a tool to determine the position of an object on the earth's surface using satellite waves. It can be used anywhere and in any weather conditions. GPS enabled device to tell the location of the device at any time.

B. *Central Software Component (CSC):*

*It* is responsible for all treatments in the central server. It is composed of an Information System (IS) MAS and security MAS.

1) *IS MAS* is consisting of:

- *Link agent:* checks if the query results are consistent with the user's profile by inspecting the privileges of the user or services requested.
- *CSC agent:* is responsible of seeking answers in the information system IS. Once the query result obtained, it will be returned back to the high level of the architecture.

*2) Security MAS:* insures the following functions:

- *Login Agent:* allows security management including the authentication and access control. Users must be authenticated by providing a username and password, to be able to own or perform actions on a component of the platform.
- *SCA (Security Certification Authority) Agent* is an agent which can attribute, renew or revoke agents digital certificates;
- *ACA (Access Control Authority) Agent* to ensure they have the access control rights permissions required for the requested actions;
- *ACC* (*Agent Communication Channel*) for exchanging critical information through a network using a secure data transmission using SSL.

## IV. USE CASE STADY

We have chosen a specific application to test and validate the proposed architecture. We implemented a multi-agent system that allows managing efficiently the WO of a Company of distribution and management of water and electricity.

### *A. Description of existing solution*

The existing solution for WOs management is formed by two components.

***Mobile application:*** allows operator to check the base of local data on the PDA that contains the day WOs, add comments, and to indicate the state of the realization of the WO (realized or need another intervention). Then after finishing the work they validate WOs realized, and send them to the central server.

***Central server application***,this component enables to manage WOs, allows schedulers to receive WOs from the services that generate WOs and assigns them later to involved operator.

The WO management process is done in the following way: The WO control center receives, analyzes and distributes WOs between different teams, involved on the ground, in such a way that each team supports WOs of a well specific sector, purpose of reducing the response delay that depends on the WO type that are either small interventions, new connections or urgent interventions, examples:

Intervention delay

| ***WO type*** | ***Intervention delay*** |
|---|---|
| electricity Interruption | 2 hours |
| Water cut | 4 hours |
| Termination | One day |
| Getting a new customer service | 1or 2 days |

Central supervisors follow the resolution of the WOs and “close” the WOs when it is solved.

The main problem with current platform is the distribution of WOs among the mobile device. Each terminal contains only the WOs for the working day. This information is loaded on the mobile device at night from a central server, through a wireless network, and the results are loaded into the database the next night, in order to generate the WOs for the next day. This system is not flexible: scheduled work for the operator cannot be changed “on the fly”, due to urgent WOs. The planner must call the operator to receive urgent WOs.

The operator must validate and sent manually the set WOs to the central server, at the day end.

Unnecessary move: at the beginning and at the end of the day the operator must move to the scheduling office in order to download or upload all its WOs for the day that lingers the work of other employees who need to check the WOs to close or reschedule for next day;

### *B. Description of the proposed WOs management platform*

The platform proposed was designed to avoid the problems of the existing solution, cited above, and to increase the effectiveness and efficiency of the field operators.

Real Time Data Synchronization Platform (RTDSP) is formed by two software components: embedded software Component (ESC) on the smartphone anther android OS and central software component (CSC) implemented on the remote central server, the communication between both components is assured by a wireless secured connection.

ESC is an application embedded in the form of a service which turns permanently tuned to new data to synchronize them with CSC. The synchronization is realized in automatic way without intervention of the user: once the PDA is switched on the synchronization is automatically established, if the ESC receives a new data a notification appears in the taskbar.

The mobile application will ask the main system periodically for WOs assigned to it from central server. The feedback information from the mobile device to the main server will be transmitted at real-time, or saved in order to be sent again if there is no communication.

ESC consumes a minimum of resources, in particular the energy and the space of storage; it is implemented as a continuous process, not visible to the user.

CSC is a web component; it has a management console which oversees equipment connected to the remote central server, posts in real time the information to the ESC and save the historic of the synchronization in a log file which is updated in real time.

The mobile devices application has to be able to communicate wirelessly with MAS running in central server. JADE LEAP works at a very high communication level, establishing TCP/IP connections between two components without caring about the physical means by which these connections are actually performed. Therefore, the application developer has to find a way of providing the connection.

RTDSP assigns WOs based on the operator location. Geolocation requires the user position data in the form of coordinates obtained from the positioning device such as *Global Positioning System* GPS. GPS is a tool to determine the position of an object on the earth's surface using satellite waves. It can be used anywhere, any time and in any weather conditions.

Android OS offers two main elements of location based service:

*1) Maps API*: Providing tools / source for Location service, application programming (API) Maps provides the facility to display, manipulate the map along with other features such as satellite view. This package is at com.google.android.maps.

*2) Location API:* Provide a location search technology used by the mobile device. Location APIs associated with the data of GPS and real-time location data. ANDROID Location API defined the android location; user location, movement / migration, as well as proximity to a particular location can be determined by Maps API.

## V. DESIGN AND MODELING OF THE ARCHITECTURE

The architecture design is elaborated by using the Agent Unified Modeling Language (AUML). AUML extends UML with the following issues [16]: a special organized agent class, the new concept of role, the new Agent Interaction Protocol Diagram.

As Fig. 3, illustrates, the agent class diagram, it shows the architecture agents and their relationship.

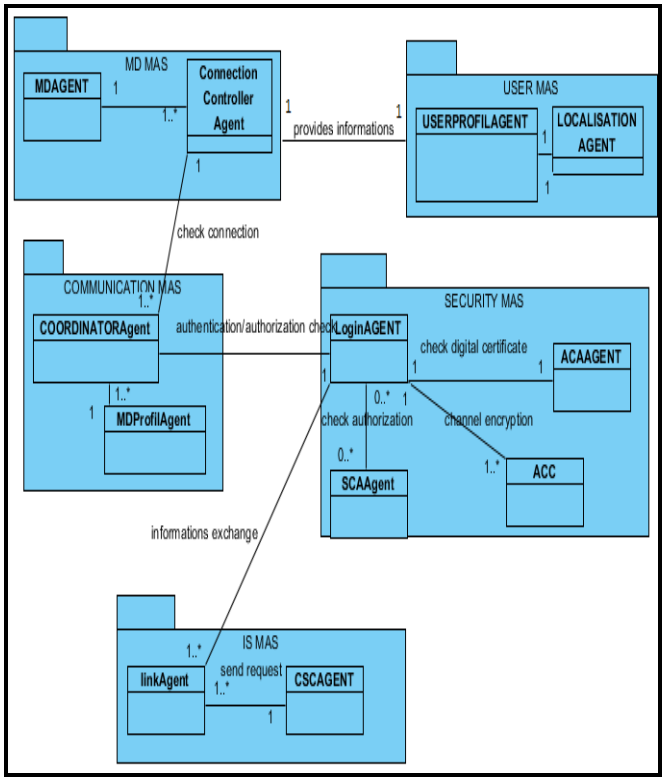


Fig. 3. Agent class diagram.

The next agent class diagram (see Fig. 4) describes the legacy of the mobile agent. It can connect to the system or disconnect voluntarily and can communicate and/or cooperate with other agents in order to accomplish individual and collective tasks. And finally, agents that run on MD can communicate through the platform [17] [18].

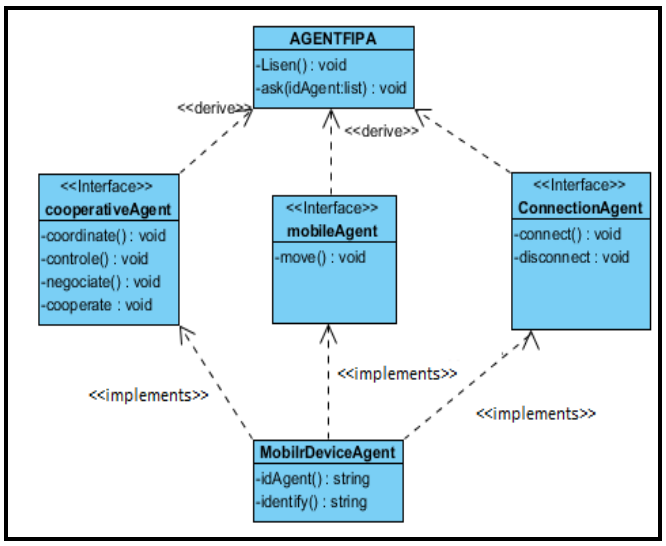


Fig. 4. Mobile Agent class diagram.

For design reasons we decompose the agent "UserProfile Agent" in two agents: User Agent and role Agent.User MAS class diagram below details the various agents (user Agent, Role Agent and Location Agent) by showing the hierarchy of the classes used for their implementation (see Fig. 5).

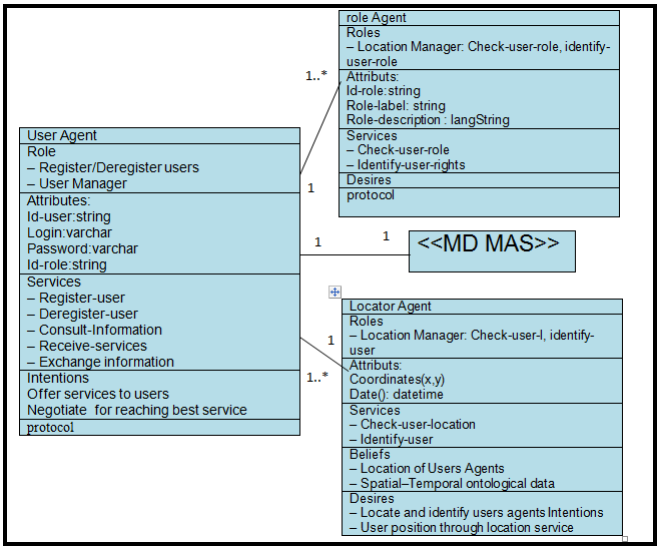


Fig. 5. User MAS class diagram.

## VI. IMPLEMENTATION

The server architecture is conventional three-tier architecture composed of presentation tier, middle tier and database tier as shown in Fig. 6. Multi-tier architecture allows separation of layers where any tier in the system can be expanded and updated with minimal or no effect on the client tiers. Each tier has a clear, focused purpose to enable high cohesion and low coupling between the tiers. The middle tier is further decomposed into business and web tier. The device manager and mobile client communicate with server's business tier over a TCP/IP and HTTP connection respectively.

On the other hand, web thin client is associated with the presentation tier of the server. The various system components which are deployed on Different machines, makes the system more modular [13].

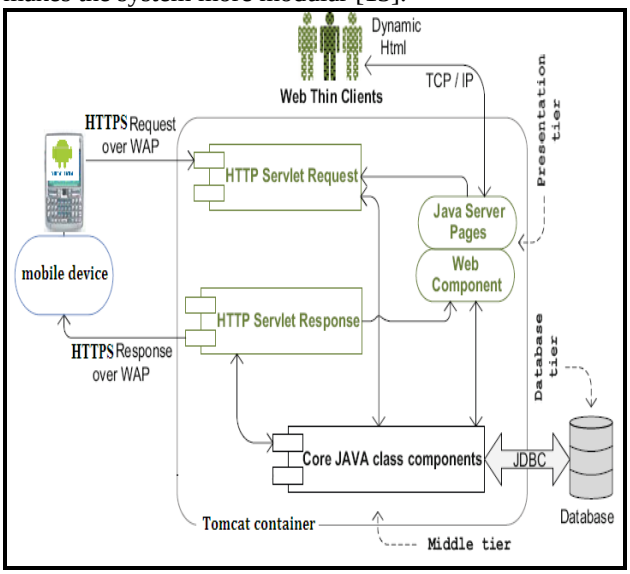


Fig. 6. Logical view of server architecture.

The ESC is implemented based on the platform Android. Android is a free mobile platform built on Linux kernel giving the authorities to customize any software layer -operating system, middleware and applications. The software development kit (SDK) is available for free to download and use and includes components such as emulator, libraries, debugging tools and source codes. Its architecture allows the developer to construct a high quality application for the best experience of the consumer [9]. Since the source code of Android was released to people, a large community of developers has organized around Android. [10] Android has a large community of developers writing applications ("apps") that extend the functionality of the devices. Developers write primarily in a customized version of Java [11]. There are currently more than 520,000 apps available for Android [12].

Android is one of the operating systems used in tablet and smartphones. There are other operating systems such as iOS, kindle, etc. However, only Android offers the following advantages:

- Allows its use in various devices, because it is open-source;
- Is in this mainstream technology branch, holding about 46.9% of users;
- Development of applications for this operating system is much easier;
- Evolution and significant growth, to an increasingly broad range of electronic devices.

In building or making Android-based applications, there are two ways. First, it must have mobile phone devices instantly Android-based. Secondly, using the emulator that has been provided by Google. Before starting to develop Android based application, it needs a few devices, such as:

- The Eclipse IDE.
- Sun's Java Development Kit (JDK).
- The Android Software Developer's Kit (SDK).
- The Android Developer Tool (ADT).
- Plug-in Eclips.

The figure below (see Fig. 7) shows graphical user interface we have achieved for a Company of Management and distribution of water and electricity.

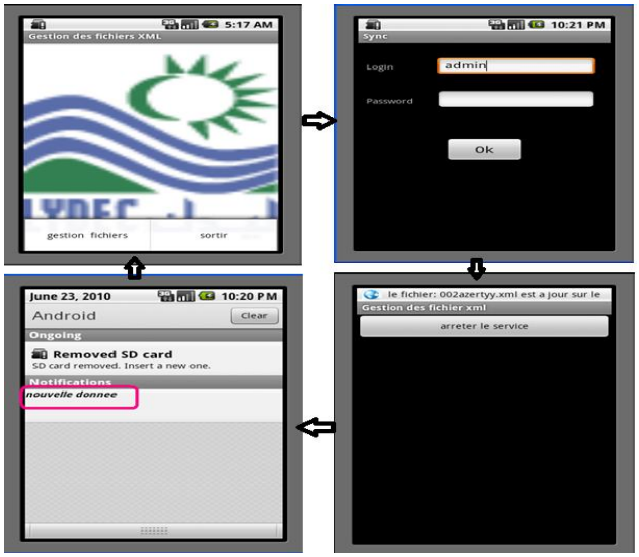


Fig. 7. Logical view of server architecture.

## VII. CONCLUSION

In this paper, we have presented Real Time Data Synchronization Platform (TSDSP), TSDSP in an intelligent platform based on a novel architecture and on MAS, able to manage efficiently the WOs of a Company of Management and distribution of water and electricity. The platform is based on two components ESC and CSC, the communication between these two components is performed using the Secure Socket Layer (SSL) protocol. TSDSP ensures the WOs exchange in real time between the company Information System (IS) and PDA of field operators, and it allows to assign WOs with the more relevant information adapted according to operator role and location. As a result, TSDSP allows optimizing and minimizing intervention time, and reduce unnecessary move to and from the company, and increasing the field operator efficiency. Our future work concerns testing our novel architecture in another case study in order to regulate and validate it.

## REFERENCES

[1] John Fox, Martin Beveridge, David Glasspool: Understanding intelligent agents: analysis and synthesis, AI Communications – IOS Press 16,pp 139–152, March 2003.
[2] J. Bradshaw, Software agents, *AAAI Press/MIT Press*, 1997,pp. 3–49.
[3] J. Ferber. Multi-Agent Systems: An Introduction to Artificial Intelligence. Addison Wesley Publishing Company, 1999.
[4] Bellifemine, F. & Caire, G. & Greenwood D.: Developing Multi-Agent Systems with JADE, John Wiley & Sons Ltd, England, 145–161 (2007)
[5] Mahmoud, Q.H.: MobiAgent: An Agent-based Approach to Wireless Information Systems. In Proc. of the 3rd Int. Bi-Conference Workshop on Agent-Oriented Information Systems, Montreal,Canada. May 28 - June 1, (2001).
[6] AgentLight - Platform for Lightweight Agents. http://www.agentlight.org
[7] MicroFIPA-OS Agent Platform. http://www.cs.helsinki.fi/group/crumpet/ mfos.
[8] Mikko Laukkanen, Agents on Mobile Devices, Sonera Corporation, (2002);
[9] Kavita Sharma , Android: The Possibilities are Endless, Singhania University, Rajasthan (INDIA) IJARCS, volume 2,n°1,2011.
[10] Maoqiang Song, Wenkuo Xiong, Xiangling Fu. Research on Architecture of Multimedia and Its Design Based on Android. International Conference on Internet Technology and Applications. Wuhan, China. 2010, p.1 - 4
[11] Shankland, Stephen, Google's Android parts ways with Java industry group. CNET News. http://www.news.com/8301-13580_3-9815495-39.html, 12 November 2007. http://www.news.com/8301-13580_3-9815495-39.html, 12 November 2007.
[12] Barra, Hugo Android: momentum, mobile and more at Google I/O. The Official Google Blog, 10 May 2011
[13] Abidar, R.; Moummadi, K.; Medromi, H. 2011. "Mobile device and Multi agent systems An Implemented platform of real time data communication and synchronization", ICMCS'11, ouarzazate, morrocco
[14] Kamal Moummadi , Rachida Abidar and Hicham Medromi (Casablanca University, Morocco) Distributed Resource Allocation: Generic Model and Solution Based on Constraint Programming and Multi-Agent System for Machine to Machine Services Volume 4, Issue 2.49-62, April-June 2012 . IGI Global. ISSN: 1937-9412, E-ISSN: 1937-9404
[15] Moummadi Kamal, Abidar Rachida, Moutaouakkil Fouad, Medromi Hicham, Network Alert Management Based on Multi Agent Systems for Surveillance and Supervising Software and Hardware Components, Vol 9, N° 6, 2014.
[16] M-P. Huget. Agent UML Class Diagrams Revisited. In Proceedings of Agent Technology and Software Engineering (AgeS), Germany, Octobre 2002
[17] Sayouti, H. Medromi, F. Elmariami, A. Belfqih, "Multi-agent Systems – Modeling, Programming and Application", International Journal of Applied Information Systems – ISSN: 2249-0868 Vol.4-N°4, April 2013.
[18] F. Moutaouakkil, A. Sayouti, H. Medromi, "Telerobotic on Internet Case Study: Mobile Robot". International Review on Computers and Software (I.RE.CO.S), Vol. 5, N. 4, July 2010.
[19] Moummadi, K.; Abidar, R.; Medromi, H. 2011. "Generic model based on constraint programming and multi-agent system for M2M services and agricultural decision support", ICMCS'11 , ouarzazate, morocco.
[20] Moummadi, K.; Abidar, R.; Medromi, "Network Alert Management Based on Multi Agent Systems for Surveillance and Supervising Software and Hardware Components". International Review on Computers and Software (I.RE.CO.S), Vol 9, No 6 June 2014.