

# Commercial Activities through Mobile Phone Distributed Processing Integrated With Mobile Agents

Meisam Hejazinia

Amirkabir University of Technology/Dept. of Computer and IT Engineering, Tehran, Iran

Email: hejazinia@aut.ac.ir

Mohammadreza Razzazi

Amirkabir University of Technology/Dept. of Computer and IT Engineering, Tehran, Iran

Email: Razzazi@aut.ac.ir

**Abstract**— The number of mobile phones is ten times more than the number of PCs today. They are ubiquitous, and provide the same processing power and memory capacity as PCs in early 90s. The mobile platform is a huge opportunity for future commercial transactions, and also many services could be conducted on them. The main challenge of this platform that hinders the development of commercial applications over it is heterogeneity in both wireless technology and the underlying hardware and software platform. Although J2ME facilitates developing application over this platform, developing application is still complex. Mobile phones should be equipped with a proper middleware that not only provides flexibility, and optimization, and usability, but also facilitates common modules for conducting commerce over them. By having such a middleware, traditional commercial applications could be redefined for this new platform. On the other hand different surveys conducted on mobile agent issues could not be applied on mobile phones due to their specific characteristics. We analyzed the mobile phone platform based commercial applications requirements together with mobile phone based mobile agents and proposed an architecture style to fill the gap, and excess the development of commercial applications integrated with mobile agents on mobile phones. We also described the existing challenges on mobile phones and the opportunities that mobile agents can provide with this new commercial activity enabled middleware on mobile phones. Different challenges are described. We provide solutions to these challenges in the form of software system including the architecture. By using the underlying society network, we will show how commercial services could be provided. Our architecture is based on mobile phone and the previous work on developing P2P computer supported collaborative work. Our contribution is providing the required mechanism to have mobile commercial applications Integrated with mobile agents on mobile phones. Our approach is a new break through to the new world of ubiquitous commercial services provided by mobile applications and mobile agents on mobile phones.

**Index Terms** – mobile phone, m-finance, m-retailing, J2ME, architecture, peer-to-peer, RMI, mobile agent, grid computing.

## I. INTRODUCTION

The number of mobile phone devices has already exceeded the number of personal computers. 4.2 billion People worldwide have mobile phones. Mobile phones have the same capability in terms of process power, and memory size as the personal computers of early 90s, yet there are not enough applications developed over this platform [2].

Current applications developed over mobile phones, follow the following architecture: The interactive mode between mobile information device and server adopts the multi-layer architecture. Considering the full sharing of enterprise information and the future extension (based on J2EE), the system adopts J2ME+J2EE architecture. It develops the server program under J2EE platform and the server end uses Servlet, JSP and EJB, etc, and connects JDBC to the backend database. As a client end, the application program on the mobile phone end is composed of various user interfaces. Additionally, some frequently used and simple datasheets with smaller data size are stored in the client database, which are connected with Web server through wireless network [8].

What we propose is, to change the perspective, and look at mobile phones as servers. Most of the work done on mobile phones until now was looking at mobile phone as terminals. The current infrastructure is used as a backbone, and mobile phones are added to this backbone. This idea is dominant in both heterogeneous wireless networks and hardware platforms, and it makes using mobile phones with limited capabilities, which always depend on the previous network and hardware/software platforms [1].

There are driving factors for using mobile phones in commerce: first, there are enormous mobile users around the world, second, most mobile phones at least support SMS and Bluetooth. In addition, there are some opportunities for 3G networks. Mobile phones are the best devices which could be used for personalization purposes.

On the other hand mobile agents are used in many applications today. We have mobile agents in grid environments, which are leveraged in load balancing [12]. In a grid environment [14], the concept of mobile agents, applied with web services, is leveraged to integrated grid

service systems. Mobile agents in a grid environment are also used in migration of services.

Mobile phones have the same characteristics of grid components, which make them different from distributed systems. Distributed systems are under the same domain policy. On the other hand, grids have different domains, with different policies. The same is true for mobile phones. They are personalized and full of personalized data. Management of this environment could not be conducted based on a centralized manager. Each mobile phone could have different policies, so it needs the same concept of virtual organizations like what we have in grids.

This mobile phone platform has some limitations, in terms of power consumption, supporting network, processing power, memory size, and input/output capacity, which should be taken into consideration when designing applications for this specific platform. There are not so many applications developed over this platform, and the main barrier was complexity. On the other hand as we previously mentioned the number of mobile phones is three times more than the number of PCs. Thus, mobile phones together could provide a huge processing power, and memory capacity. In order to leverage these capacities, distributed mechanisms should be developed. These mechanisms could not be the same as what we have in a grid environment. Grid computing has a high processing power, and a high memory capacity, but here mobile phones do not have these facilities. Moreover, input and screen capabilities are heterogeneous.

While grids consist of heterogeneous devices, this challenge becomes more severe on mobile phones. We not only have device heterogeneity, but we also have network heterogeneity, that should be handled. Mobile phones mostly support Bluetooth, and some support WLAN. So we do not have a seamless network like grids. Since mobile network is heterogeneous they are imposed by high network failure.

Mobile agents are autonomous processes, which collaborate with each other, and change the dynamic environment. Mobile agents are used in resource management, workflow management, and performance optimization in grids [12]. Mobile agents have four characteristics. Firstly, they have autonomy and computing capabilities such as learning and reasoning which are determined by the agent itself. Secondly, they are adaptable, which means they adapt their behavior according to the dynamic environment. Thirdly, they have interactions, which mean they affect other agents, and ask them to do actions via messages. Fourthly, agents are interoperable, which means they can perform their own jobs and cooperate with other agents to achieve their goals.

Mobile agents improve grids' performance, by grid resource learning and determination of interactive agents. The main agents' advantage is that they provide flexibility while they are autonomous. Different approaches for having mobile agents on mobile grids are proposed [12][14]. These approaches mainly use a block

called agent management system, which monitors and schedules agents, and provides them with management mechanisms such as create and delete. These approaches also use a hierarchical directory facilitator, which act like yellow pages for agents. These approaches could not be used on mobile phone platforms, since they have different characteristics from grids, as we mentioned previously.

Any software system, on mobile phones, for their unique characteristics, should have different nonfunctional requirements. This system should be lightweight and optimized for a lower memory capacity and a lower processing power, and battery power that are the most critical attribute and it must be preserved. Since mobile phones are heterogeneous, adaptability is a very important nonfunctional requirement. Flexibility is also desired for systems on mobile phones. These entire nonfunctional requirements are considered in our design.

By a quick look at these non-functional requirements, we find them contrasting, but if we look into details we find out that this problem is the same as other optimization problems. So we should change our perspective and look globally and as a result, the problem could be solved, if we just provide flexibility in terms of distributed processing, and provide optimization locally for CPU and memory usage, and globally by reducing the number of messages transferred between mobile phones, and roam between different choices of networks available. For robustness, we need to have good exception handling, and proper protocols for distribution handling.

Because mobile phone devices lack standard application platforms and operating systems, it brings the application programs developed for mobile communication devices, higher risks of interoperability. To solve this problem, J2ME provides a proper software layer, but still it is complex, and as a result of that there are not many commercial mobile applications over it. The second problem is that still network programming is complex, especially for Bluetooth, with the odd concept of master and slave. The third problem is the lack of RMI support; you cannot invoke a method of another object on another mobile phone, which is the key point in commercial applications development.

J2ME adopts the modular architecture composed of JAVA virtual machine layer, configuration layer, profile layer and MIDP layer to meet the demand of developers. J2ME consists of four layers, the first layer is JVM, which is customized for the host specific J2ME configuration, while applications rarely see this layer. The second layer over JVM, is the configuration layer, which defines the function of JVM and smallest set of Java Class library available on specific class devices. Developers can support these functional units and libraries available on all devices belonging to a certain specific class.

J2ME has two configurations defined by Java Specification: CLDC (Connected Limited Device Configuration) and CDC (Connected Device Configuration). The first one aims at the limited devices and the size of memory is often between 128KB and 512KB. The second aimed at the devices with higher

configuration and these devices may have 32-bit or 64-bit processors, and at least have the storage space of 512K. CDC uses virtual machine called CVM. While the system based on CLDC uses virtual machine KVM. Users rarely see this layer but it is very important for the realization of profile. The third layer over Configuration is the Profile layer, which defines the smallest set of API available on specific series devices. MIDP is the mobile information device profile layer. It is a Java API set, which mainly deals with the problems, such as user interface, persistent storage, networking, and etc [8].

The implication [3] shows us these complexities are common in most commercial applications over mobile phones; also these complexities could not be solved by normal developers. There are two main challenges for mobile phone platforms: first, is the lack of a suitable business model, and second, is the lack of a suitable middleware for supporting these key issues. These challenges made the mobile platform unsuitable for development, and we cannot see enough commercial applications purely on mobile phones.

The solution for these problems we proposed is an architecture style together with software mechanisms which makes this platform suitable for commercial application development integrated with agents. A mobile platform is a huge opportunity for future commercial transactions, but this platform should be equipped with a proper middleware, and then traditional commercial applications could be redefined for this new platform.

The paper is organized as follows: Section 2, describes different proposed commercial scenarios that could be redefined for this new perspective of mobile phone commerce, also it discusses scenarios and requirement of mobile agents on mobile phones which could be integrated with this commercial scenarios. Section 3 provides a definition for society network and explains why it is suitable for mobile platforms, it also describes different network infrastructures, for developing commercial applications on mobile phones; and proposes a solution for heterogeneity. Section 4, provides proposes a middleware architecture style for commercial application development over mobile phones together with our solution for mobile agents on mobile phone and provides details on how differently the challenges should be overcome, and Section 5, provides the concept of process granularity of commercial applications over mobile phones it also discusses the evaluation issue related to our proposed solution, and section 6, concludes the paper with highlighting our contribution.

## II. COMMERCIAL SCENARIOS

Let us start with a scenario; Suppose John has an application on his mobile phone recommending him to choose the portfolio for the stock market. This application is The Intelligent Risk Management application that not only needs the global risk information, which is real-time and is in different places, but also it needs complex calculations. His mobile phone has some limitations in the dimension of the processing power, and memory size. But John is with his friends, and his friends are located in

different social networks. John cannot open his laptop and wait for its start up and then connects to the internet, while he is in the street. Thus, if he could use these social networks and his friends let him use the unused mobile phone processing power, memory, network, and battery capacity, he is able to run his application, by just distributing it over his friends' mobile phones. Each mobile phone in this scenario has its own tasks that is locally initiated or initiated by other mobile phones in the same social network with it. In this way, life becomes easy.

This scenario has one important requirement, which is the underlying middleware. This middleware should provide security, and support RMI for invoking methods on other mobile phones. It should support protocols to distribute tasks; it should break a task into parts and assign it to other mobile phones. In this way, mobile phones together will be converted to a super computer. Moreover, it will transparently handle the job that is assigned by a special application on the software layer.

Mobile phones are on most of the time, while PCs are sometimes off. Mobile phones are most of the time in the proximity of each other in a social network, but PCs are connected to each other through the network infrastructure. The main point is that, most mobile phones do not support WLAN and 3G due to being expensive for people. Thus, it is not easy to connect to PCs to assign jobs to. With these specifications, the solution is to use other mobile phones' processing power and memory capacity, which are always free, and wasted.

This scenario could be extended to scenarios for portfolio optimization, mobile payment, and broader to mobile retailing [4].

The second scenario is about mobile retailing [3]. We could have different people having different products, with their specifications and prices, there are also different people having interest in buying products. Then by broadcasting the request over mobile phones, using experience of people with different credits, the person could find proper retailer to buy the product. These scenarios could be extended by personalization, and location based service, which are the unique attributes of the mobile phone platform. The whole scenario could be done during a day, when a person is going to work, during the work hours and even when he is coming back home. Even we could go further and have deals and contracts signed over mobile phones, and this entire scenario is done by simple data transfer over the underlying middleware, which is recommended by our paper.

The third scenario is about credit financing [9]. Effective information sharing, risk mitigation and cost reduction are the most critical challenges to encourage financial institutions to finance individuals and firms. According to an extensive study, underlying reasons included for problems in financing is that there is an information asymmetry between financial institution and individuals, high risk scoring of SMB induced by lacking good credit history and qualified deposit, high transaction cost to profit ratio. Financing requests arising from various phases differing in the degree of risks. If financiers

could have comprehensive information about the purchase orders, account receivables, invoices, and individual's social network, then credit financing could be done easily. Many people are not engaged in e-commerce transactions, so information from e-commerce systems is incomplete. Mobile phones have a big advantage that is portability; they are with you anywhere and anytime. Thus, their simplicity makes more people to use them in order to do their commercial transactions, if there are suitable commercial applications developed over them. In this case, the information related to people's transaction's history, could be comprehensively stored over their mobile phones. There is also the other more important possibility, which is using people's social network. Mobile phones contain people's calls duration, and number, so with this, the financier could use this information with the user permission, to know people's credits related to the social network. Risk control is the core of any financial services and business models, and our middleware could provide the risk control. Our model provides a platform to develop innovative financial services to address the issue. Now let us discuss different scenarios which mobile agents could be integrated in this platform.

Grids are used for scientific, financial, commercial application which needs high capacity, and large processing power. But such applications could not be used on mobile phones. Mobile phones are portable and personalized, that is the key for providing mobile agents on mobile phones.

The first question is why should we have mobile agents on mobile phones, while we could have static services? The second key to answer this question is the battery power, which is critical in mobile phones. If we have a consulting application, which will guide you to a decision, through answering some questions, we have two approaches to handle it. First, we could use services on mobile phones, which are static. Then, you may have 10 questions, and after answering each one, you may ask different kinds of questions according to the answers. This means the next question depends on the answer of the previous question. This is the same as the 20 questions game. If we use this approach of static services, 20 messages are sent and received by the provider and consumer of service. This leads to a high battery usage, which is not desired on mobile phones.

The solution to the previous scenario is mobile agents. We could describe each mobile agent as an autonomous process; they are simple codes, which could be executed in a container. These codes have logic; we call that logic the decision graph (like decision tree). By sending a consultant agent to the consumer, our mobile phone does not need to send and receive 20 messages; instead it only sends 5 messages and which means 75% save in the battery usage of mobile phones.

The supporting scenario to use mobile agents on mobile phones is not limited to a simple consulting. However, the consulting scenario could be defined more broadly. For example, you may be a financial consultant, and want to recommend your customer which of eight

options of stocks to use. In this scenario, you do not want to let the customer know your logic of decision making, since it is your core competency.

We could have other scenarios. For example, you are a seller of eight brands that are for one type of product. What you do is you go to a person, and ask them the requirements, and suggest which product to choose. All the attributes of the previous scenario are the same and only the context has changed. We could have many different scenarios related to doctors and travel advisors. Anything that contains decision making and choosing between options is included in this scenario, and could be provided by mobile agents on mobile phones. Agents should not always interact with people themselves. People could have their own agents on their mobile phones. In this way, the agent who migrates to their mobile phones will communicate, with the local agent, and this will make communication automatic. The main advantage that supports this scenario only on mobile phones is that mobile phones are personal. People put their personal profile on their mobile phone, and their local agent could use this personal data to negotiate and communicate to answer question of mobile agents, which act as advisors.

In our scenario, communication means exchanging the information that the mobile agent needs. Perceptions and reasoning is done by getting the required information from the local agent. Using agents on mobile phones provides advantages of load balancing. In this way agents could roam on different mobile phones on the society network, which we will explain in the next section. Using agents on mobile phone will reduce the battery usage, which is very significant in mobile phone networks. Let us look into what a society network is. In section 4 we will discuss thoroughly the advantages of providing mobile agents on mobile phones.

### III. NETWORK INFRASTRUCTURE

There are four options available for providing a middleware, over mobile phones: Bluetooth, SMS, WLAN, and GPRS. GPRS is based on GSM. In the middle of these two technologies is the improved version of GSM, which cellular operators enhance their systems, to improve to GPRS, and SMS is built over GSM, it is simple, fast highly flexible, scalable, wide spread and user friendly. The specification of each of these infrastructures could be seen in figure 1.

To make decisions over these infrastructures, Bluetooth shows up, for it is free but has some

TABLE I.  
NETWORK MEDIUM COMPARISON

	Bluetooth	WLAN	GPRS
Bandwidth	1 Mbps	11Mbps	115-117 kbps
Power	1-10mw	50-70mw	200-800mw
Range	10-100m	100-200m	1KM
Cost	None	Low	High
Frequency	2.4GHz	2.4GHz	900/1800MHz

shortcomings upon its limited range. Bluetooth and WLAN support the concepts of same time and same place. At first sight, we may think that this would not be a suitable infrastructure to conduct commerce. But when we look thoroughly, we find out that there is another thing, which is society network. Society network means people we meet every day, in buses or bank queues, waiting for our flights. We are not staying in a place day and night, and we are moving, and this is a great opportunity for providing exchange with many people. Although we meet and talk with limited people, our device is not limited like us. It could automatically interact with many people we may not see, but they are in our proximity, and it could match profiles, use their processing power, or memory. Our device could even trigger and guide us to the right person. With this mentality, Bluetooth is not limited anymore, while we are not Robinson Crusoe who don't have any people around us in an isolated island, even when we are in an island there are many people around us, and this means a huge opportunity for computing and commerce. The main advantage of Bluetooth is that, we are not controlled by central agents anymore, there is no operator, and we could conduct commerce without the middleman (operators).

After that we have SMS, which is accessible from anywhere, but has a shortcoming in terms of cost and middleman controls and regulations. The third suitable choice is WLAN which is also free and suitable, with the ability to connect to the internet, and the fourth one is GPRS, due to having a high cost of transfer; it makes it difficult for a normal application to communicate.

To overcome the shortcomings of each of these options there is a solution, which is a software layer over them for roaming [6] transparently, so that the application over them wouldn't be notified that the underlying layer has changed the network medium. Thus, the first underlying layer searches for the mobile phones in the proximity to check whether they have the special service. If they do not, the software layer under it does the roaming and goes to WLAN or SMS according to the user preference between cost and time. In addition, if it is not found, then automatically it connects via GPRS to another mobile phone, or sends an SMS to a default mobile phone that he knows it contains the required service.

#### IV. ARCHITECTURE FOR CONDUCTING COMMERCE OVER MOBILE PHONES

So far, we have discussed main issues for the enhancement of commerce on mobile phones. The main challenge today for developing applications over mobile phones is that, there are not enough software infrastructures, for distributed processing, specially RMI support, over mobile phones, fulfilling four qualities, usefulness, performance, flexibility, and robustness simultaneously. For solving this problem you can see our proposed architecture in figure 2. We are going to describe the components.

This Architecture is an enhancement of the previous architecture of Peer2me [7] framework, which was developed over mobile phones for collaborative work. We

enhanced it so that it could support the distributed processing and commercial basic functionalities required for conducting commerce over mobile phones. The Domain layer contains an abstraction of concepts for network modules, it contains, node, group classes.

The Network Module layer is an implementation of network specific classes and methods. The Network Interface layer separates the implementation layer from the application layer, to decouple application from the underlying network implementation.

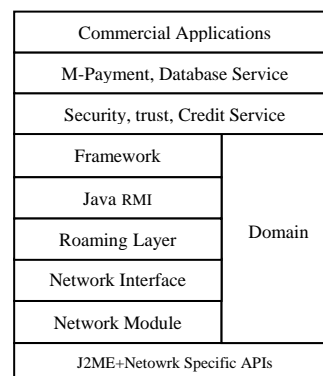


Figure 1. Proposed Architecture for mobile phone commerce

The Roaming layers, provide transparency for applications, so if the user was moving, and he exits the coverage of one network medium, then automatically this layer switches the network medium, according to the user's preference, without interrupting the application.

By this architecture, we have different peers for supporting commerce, and each peer provides different services. We should have bank peers, the mobile phones which support the storing account information, and do transactions. Also, we could have some directory peers, the mobile phones that provide addresses of the bank peers. We could have authentication peers, which are mobile phones that provide certification of the special buyer peer, or seller peer, and this service could be provided with SMS, for the security attribute.

The Security, trust, Credit service layer are the basics for any commercial applications. Even for payments, in this layer, people rate each other, and specific credits. Trust algorithm could be deployed. For example, it could be a simple rating or a complex one, which incrementally corrects itself by multiplying the credits of different people and chooses between them. The point is that, all these interactions are done using the underlying layer of network, which is decoupled from the network implementation, and even we could define a policy in our Roaming Layer, in which this service is being done by SMS and using a specific database or by Bluetooth, using special hash codes.

There are different types of payment, (1) Petty Cash Payment: The amount of money paid below 2 Euros. (2) Small Payment: The amount of money paid between 2 Euros and 25 Euros. (3) Large Payment: The amount of money paid above 25 Euros, it can be regarded as Acer Payment as well. By this architecture even Large

Payments could be supported. If different peers provide credits for each other, according to the game theory we will not do coalition with everyone. In this way, if and only if we do successful transactions with many people, we get credits from them. If we restrict payments to small payments for each individual, we get credits from. Then, we could have Large Payment if and only if we get many credits from many peers, and this will solve the problem of large payment over mobile phones.

The main advantage of our architecture is that we decoupled network from other higher layers, and one can easily develop network for new network technologies. The most important point is that, one could even have payment without the existence of a middleman like an operator. Instead, you are dependent on the society, for they provide you the credits. By the credit you get from the society you could have commercial transactions using your mobile phone.

We propose also database services, in which we recommend new developing concepts of micro databases that data are distributed over different peers. Each peer, itself does optimization on itself according to its policy, and utilization function and what information to put on the current device. Although the general recommendation policy, as a default could be developed based on security, trust, and credit service layer, still each mobile phone is autonomous, and could decide whether to put some data on its storage or not. Each mobile phone would provide database services according to their own databases (e.g. providing free experience about a specific product).

To support this architecture, we need to have the unique address of each mobile phone, which could be the phone number. Also for providing a more dynamic atmosphere, we could provide roaming on the upper layer, which means if someone changed their SIMcard, then we could have some peers, for providing the directory service, and this service could be provided on the roaming layer. Now it is time to discuss about how to integrate mobile agents into this platform.

Mobile agents in our software system contain two parts: the first part is the logic of the agent. This part could be roamed in society network; it could be broadcasted to the other mobile phones in approximation as the service provider is walking. Moreover, mobile phones, who wish to advertise this service, could have this part of mobile agent on their mobile phones to advertise it for people who want it. The second part of mobile agents is the container that is built in the middleware layer; each logic of the agent could be executed using this container. Each mobile phone has a local agent to negotiate with other mobile agents. To communicate between agents, we need some protocols. We could have different protocols and conventions. For each convention we have a CID that is defined as metadata of each agent. To communicate through a special convention we have a convention service. Convention could be loaded from the mobile phone that transfers the agent, or it could be the response of a request of convention from other mobile phones that inputs CID.

Each mobile phone has a special policy. Providing data from the profile is done by the local agent according to that policy. Also this policy defines whether a person is willing to host special mobile agents or not. Each mobile phone has a queuing service. Mobile phones put the incoming preferred agent in the queue. If the person is willing to host a mobile agent, the agent stands in the queue. How to put agent in queues is defined in local policy of user. Each agent has a source address, which is defined as UID. Additionally, it has a unified identifier which contains the code of the agent, which is AID concatenated with UID. Each mobile phone could have a cache and put the favorite agent's identifier in it, and ask from the agent in the future, when they will want it again. Each agent has a specific life time, and it will die upon the finishing time. This is useful for agents who may update. Each agent has a specific credit, which is calculated using people's rate put on it. This credit is updated when the agent serves a specific person, and it is calculated by the average rate, people have so far put on them. This credit could be used by the person, whom this agent is offered to. A person could define a local policy so that, it would not accept agents with a rate lower than 0.6. This policy is executed by the door keeper module in the mobile phone. Thus, the credit and rating of each mobile phone are done through a number between zero and one.

Each agent could have a cost. Any time a person wants to use it, could pay for the agent's use by m-payment. They could advertise an agent, so it could earn money by that. Each time a person uses a mobile agent, contacts to the center to pay. The center could know who has advertised this service and pays for the advertisement in the same model as the model per click on search engines is paid. One user may set special policies relative to the lifetime, cost, and credit of the agent to let it in. When agents are hosted on a mobile phone, the mobile phone could do three things: it could prevent agents to come in, it could advertise agents, or it could use agents, so agents will be put in the queue. User could define special policies to kill the agents, for example it could be by time, credit or other policies.

The main advantage of this model is that, if service that one agent has proposed, is not valuable, it will be killed, and so we wouldn't have congestion in broadcasting an agent. User may also define policies to broadcast this agent to an only person who is their friend.

Each agent could have a special policy to prevent people to hack its logic. For example, it could be defined to not roam over more than 5 levels, or to not let local agents to communicate more than a certain time. In addition, different pricing policies could be adopted for agents inside it. This pricing policy could be used by the pricing container on the host mobile phone. Each agent could have a special logic for learning. Learning of an agent could be divided into two parts: the first one is local learning, in which logic will change according to the user's input, in other word tree of choices changes, according to user input, by agent. Secondly, we have central learning; in this case, agent will send back the

feedback it receives to the UID, who creates it. In this learning, the creator will change logic, and provide a new version of the mobile agent. This feedback provided by the agent to the creator could be prevented by the mobile phone's door keeper according to the set policy.

The convention, which defines how local agents speak to mobile agents could also be cached locally on mobile phones according to the ordinary convention is used. All pervious activities are conducted in the middleware layer. We discussed cache previously in this section. But the point is that capacity of cache could be configured according to the local policy. This functionality will help us provide adaptability on mobile phones. The point we should consider is that agents should be lightweight that is very important. So consultation over a limited number of choices should not be implemented in the agent any more. If a person wants to consult someone on more than one choice, they could have different agents hosted on their mobile phone. In this way, it will preserve the result of consultation with each agent, and compare them with each other. Thus, he, himself, could choose between them. This decision is done, while processing on a single machine consumes much less than the network battery power. So the user may have local applications, or local agents, which aggregate the result of a number of agents' consultation, and will provide users with the result. This lightweight agent could also be transferred over a limited number of SMSs.

What we discussed about method of agent distribution was all push models, but if someone wants to pull an agent for the first time what should they do? The search service is not a big deal for grids, but when it comes to mobile phone platforms, it is really challenging, while mobile phones do not have enough processing power, and they do not access the huge metadata related to the agents and conventions. Even if we want to have a simple search over mobile phones, the input capability is very limited, and it makes difficult to search over them. But what is the solution for mobile phone? If we cannot have search, then there will not be any data grid over mobile phones.

We presume we should use the PC platform to provide search, but on mobile phones, each agent has a specific AID concatenated with UID and each conventions has specific a CID as we explained before. We leverage the PC platform to find these identifiers. Since agents and conventions we are searching for could be repeated over time, then simple map of domain, especially ontology these identifiers, could solve our problem. This mapping is between metadata to identifiers by using hash functions. If a person does not have enough time to go to PC, what should they do? We could have two solutions: firstly, a centralized service, which you can contribute by SMS, then that center, there could be an automatic machine, which according to your history of preferences helps you, or it could be a person behind SMS system, or you can use the call center to find out identifiers. The point is that using web applications configured for mobile phones, is not desirable on mobile phones, since it has limited input abilities, and a small screen, so we recommend using our two solutions. Figure 2, shows the

architecture of our software system to provide mobile agents on mobile phones, the relations between components is explained in detail in the text.

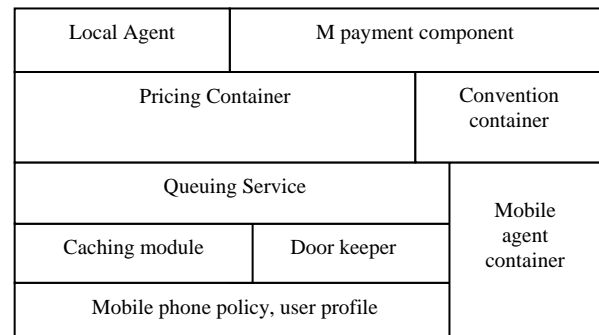


Figure 2. Software system proposed for providing mobile agents on mobile phones

Our proposed software system helps load balancing, which is very critical in mobile phones. While unlike servers, they could not handle a high number of requests for a limited memory capacity and processing power. In agent based services, which we propose anyone needs services, will host the agent to serve him/her, and we do not have such load on special mobile phone. The other advantage of having mobile agents on mobile phones like our system is that by sending the credit to the center each time the agent is used, we will find out which service is popular in a special place. It is somehow a marketing tool to find customers and expand markets. Agents could be stateful but they should be lightweight, and being stateful helps them to learn. Our approach does not have a central agent management system, which exists in grid, and we have no single point of failure, of course we could not have them on mobile phones.

## V. THE PROCESS GRANULARITY CONCEPT

Previously we discussed having two simultaneous non-functional requirements, which were flexibility, and optimization. At first glance, we find it contrasting, because, if we want to have a flexible application, it would have some overhead, which contradicts optimization.

Mobile phones for their capacity constraint need optimized applications. But if we look globally, there are many mobile phones that do not use their infinite capacities.

These specifications, guide us to the concept of process granularity. Previously, the concept of granularity was developed in databases, but today, for this feature of mobile phones, we should inject it to the concept of process(literature). If we decouple data from logic and break the processes to many simpler processes and distribute them, each of these simpler processes also could be broken into pieces and distributed over other mobile phones. After that, we could collect the results, and make the required results in each level.

The main point of this approach is to decouple the workflow of the special commercial application, from each process in every level of breakdown.

With this approach there wouldn't be any limitations on mobile phone platforms. Moreover, we should use some optimization algorithms with the required time, available battery power, and estimated level of breakdown, network, processing, and memory power usage, to find out whether we should do this sub process locally or send it to other mobile phones. There are situations, where we cannot do it locally, while we do not have that method, we should use RMI capabilities that are supported by our proposed middleware.

New protocols should be developed for these concepts that are configured according to the special requirements of this platform. Also economical system should be developed over it, accordingly, and the costs are input of the optimization algorithm.

To evaluate our system, we should first have other systems to compare with. Unfortunately there is no such system to compare with, and also previous works on grid were not suitable for this platform, for unique characteristics of this platform. But we compare the proposed system with nonfunctional requirement of this specific platform.

First being lightweight, is the main nonfunctional requirement, it is supported by our layered architecture. Modularity of proposed system provides flexibility, and adaptability. This modular system, could provide even lighteweightness, which means someone who has a mobile phone with a higher memory capacity and processing, can install more modules, and enjoy optimized applications, and less battery power usage. Our proposed system also provides adaptability, that means the threshold of caches, and other parameters could be set according to special devices requirements.

Our mechanisms are optimized and less computational, and more distribution makes it more suitable for mobile phone platforms. Also our platform is scalable, since our mechanisms are configured for distribution, and do not depend on the central component. The main characteristic as we talked before is flexibility that mobile agents provide the system with. The other characteristic is load balancing, which is also provided by our proposed system, as we explained before.

## VI. CONCLUSION

Mobile phones provide a very good opportunity, for commercial application development, for its ubiquity, which means you can use it real time in any places, accessibility, which means you have the freedom to use them anywhere, anytime, security, SIMcard are smart cards that contain confidential users' information, convenience, for its hand held size, and light weight, and personalization, which means it is not shared between users and could be adjusted for users' needs.

Unfortunately, there is no suitable middleware to support commercial activities purely on mobile phones that addresses the special constraints of limited computing power, limited information display for navigation, limited

power and small memory size. Also there is no software system to provide mobile agents on mobile phones. Most previous works were focused on looking at mobile phone as terminal device. We changed our view, and look at mobile phone as server, we defined concept of society network, which could be leveraged to enhance mobile phone mobile agent capability. Our contribution was redefining mechanisms of mobile agents for specific characteristics of mobile phone. Due to unique characteristics of mobile phone, previous solutions were not suitable, and we defined the mechanism ourselves. Our publish agents services was not done globally, like grid, but we used locally publishing, we used lightweight agent concept. Our approach was adaptable, and provides services according to local policies. We used caches, to improve performance.

We also proposed an architecture style for the required middleware, with the goal of usefulness, performance, robustness and flexibility. Additionally, we introduced the concept of process granularity over this new platform that could support scenarios we described in section 2. Our architecture provides flexibility, adaptability, and performance for mobile phones. We identified new scenarios for using mobile phone for mobile agents, which was focused on consultancy services. We talked about mechanism needed for communication, which are configured for specific characteristics of mobile phones, and we abstracted protocol from agent, and agent from its container.

There are still challenges related to optimization of algorithms, and economical systems which are properly configured for this platform, and could be the subject of future research. In addition our work could be enhancing by investigating threshold for different mobile phones for cache. Also simulating our framework and investigate how granular each agent should be, could be issued for future investigation.

## ACKNOWLEDGMENT

This research was created through a financial grant offered by the Iran Telecommunication Research Center.

## REFERENCES

- [1] W. Niu, L. Bai, "Business-based Enterprise M-Commerce Models", IEEE 2008.
- [2] D. J. Xu, S.S. Liao, Q. Li, "Combining empirical experimentation and modeling techniques: A design research approach for personalized mobile advertising applications", *Decision Support Systems* 44.pp. 710-724, 2008.
- [3] C.K. Georgiadis, S.H. Stergiopoulou, "Mobile Commerce Application Development: Implementing Personalized Services", *7th International Conference on Mobile Business*, IEEE2008.
- [4] J.Hu,N.Zhong, "Developing Mining-grid Centric e-Finance Portal", *Proc. of the IEEE/WIC/ACM International Conference*, 2006.
- [5] X.Yang, Z.Qiang, X.Zhao, Zh.Ling, "Research on Distributed E-Commerce System Architecture", IEEE 2007.
- [6] C.K.Chen, C.W.Chen, C.T. Ko, J.Lee, J.C.Chen, "Mobile Java RMI support over heterogeneous wireless networks: A



- case study”, *Journal of Parallel and Distributed Computing*, 2008.
- [7] A.I.Wang, M.S.Norum, “A Peer-to-Peer Framework for mobile collaboration”. *International Symposium of Collaborative Technologies and Systems*, 2007.
  - [8] X.Zhang ,”Design of mobile electronic commerce system based on J2ME”, *proc. International Conference on Electronic Computer Technology, IEEE*. 2009.
  - [9] W.SUN, J.LI, D.LIU, G.FENG, P.LI, “Business Models and Solution Architectures for SMB Financing in a Supply Chain Ecosystem”, *Proc. IEEE International Conference on E-Commerce Technology for Dynamic E-Business*, 2004.
  - [10] X.Wang,F.Liu, R.Yang, F.Xie, “Social Computing-Based Trust Establishment in E-Commerce”, *Proc. International Conference on Internet Computing in Science and Engineering, IEEE*, 2008.
  - [11] L.Yunhong, L.Siwen, “Research on Mobile Payment in the E-Commerce”, *Proc. International Conference on Management of e-Commerce and e-Government, IEEE*, 2008.
  - [12] M.Bunruangses, W.Poompattanapong, P. Banyatnoparat, B.Piyatamrong , QoS Multi-Agent Applied for Grid Service Management, *Proc. ACM*, 2004.
  - [13] T.E. Athanaileas, N.D. Tselikas G.V. Tsoulos, D.I. Kaklamani, “An Agent-Based Framework for Integrating Mobility into Grid Services”, *Proc.ACM*, 2008.
  - [14] F.Magoules, J.Pan, K.A.Tan, A.Kumar, “Introduction to grid computing”, Taylor & Francis Group, 2009.
  - [15] C. Blundo, E.D.Cistofaro, A Bluetooth-based JXME infrastructure, *Proceedings of the 9th International Symposium on Distributed Objects, Middleware, and Applications (DOA'07), LNCS*. 2007.