Volume 51, Number 1, 2010

ACTA TECHNICA NAPOCENSIS
Electronics and Telecommunications

A STRUCTURE FOR HARD HANDOVER ANALYSIS USING AGENTS TECHNOLOGY

Gabriel SIRBU[1], Ion BOGDAN[2]
[1] *Dunarea de Jos University of Galati, Electronics and Telecommunications Dept.*
Domneasca-47 Str., Galati-800008, Romania, email gabriel.sirbu@ugal.ro
[2] *Gheorghe Asachi Technical University of Iasi, Telecommunications Dept.*
Blvd. Carol I - 11, Iasi-700506, Romania, email bogdani@etc.tuiasi.ro

Abstract: **The mobility of the users in a cellular mobile environment is involving the use of the radio resources. The main aspect, which characterizes this attribute, is the hard handover and this must be carried out without blocking the existing calls of the static positioned users. Combining the two sets of users, namely static positioned users and mobile users, the resulting system must be able to manage the conflict situations, resulting in a minimum blocking probability, achieved with a centralized structure, or a distributed structure. Our work is introducing a minimal distributed structure that can be used to analyze the behavior of a cellular mobile system, based on the agents technology. The structure includes two agents and each of them implements the entire operations specific to a dynamic channel allocation scheme, into a cell. The third agent is added in order to manage the eventual hard handovers requests. This three agent minimal structure can be then multiplied in order to extend to, to simulate and to analyze various real conditions.**

Keywords: *cellular communications, hard handover, user mobility, channel management, Dynamic Channel Allocation Scheme, cost function, MASE technology, fixed agents, blocked call probability, forced call probability.*

I. INTRODUCTION

A. CELLULAR MOBILE ENVIRONMENT - HANDOVER

The handover mechanism is specific only to cellular systems, characterized by users mobility. There are many techniques involved in this mechanism, having the goal to reduce the blocking probability for the on going calls of the users which transit the radio cells bounds. These techniques are providing the reserved channels for handover situations, and they are introduced in [1] as Prioritized Channel Assignment. Also, these techniques can be applied on Fixed Channel Allocations Schemes, as well as to Dynamic Channel Allocation Schemes. In our work we analyzed the traffic in two cells, each cell having the Dynamic Channel Allocation Scheme management for static positioned users. Each cell can provide each other a mobile traffic, randomly generated, on a carrier to interference plus noise ratio – CINR basis. An agent analyzes the mobile traffic and makes the management of the radio channel resources associated to this traffic.

B. DYNAMIC CHANNEL ALLOCATION SCHEME

Channel Allocation Schemes are techniques meant to resolve the conflicts between multiple carriers in radio communications systems. In our work we implemented Dynamic Channel Allocation Scheme, which consists in splitting the service area into cells and giving each cell the permission to use a set of radio channel, on a carrier to interference plus noise ratio calculus basis [2]. We used expression (1) to compute the CINR where N is the thermal noise, P_0 and P_i are the transmitting powers of the users u_0 and u_i respectively, α is the path loss exponent, ξ_0 and ξ_I are the standard deviation of the log-normal fading (shadowing) associated to the users u_0 and u_i respectively and A is a network specific propagation coefficient.

$$R_{cni} = \frac{A \cdot P_0 \cdot d_0^{-\alpha} \cdot 10^{\frac{\xi_0}{10}}}{N + \sum_{i=1}^{m} A \cdot P_i \cdot d_i^{-\alpha} \cdot 10^{\frac{\xi_i}{10}}} \qquad (1)$$

In order to calculate the CINR we considered a 21-cell pattern, described in [6] and we integrated this pattern into an agent. Finally the resulting structure contains two agents, each agent providing all the specific operations into a cell: new calls management and radio channels management.

C. AGENTS TECHNOLOGY – MASE TECHNOLOGY

From the structure point of view, a software agent is an entity created with a final mission: to execute different tasks in order to achieve a specific goal [4]. From the mobility point of view there are two types of agents, fixed and mobile [3]. Fixed agents cannot move between entities that create the framework of the structure. In contrast, mobile agents can migrate between entities of the framework in order to achieve their specific goals. As it is described in [3], the agent structure needs several items to be implemented, in order to achieve the final goal:

a) Agent Platform – this is an environment in which the agent can be deployed; we used the AgentTool platform, version 1.8.3. AgentTool is an automated platform implemented in Java and with the aim of agentMom it provides the basic building blocks for agents, conversations between agents and the messages that are passed through these conversations.

Manuscript received January 10, 2010; revised March 23, 2010

b) Agent Communication Channel – this allows agents to exchange information between one another concerning communication messages; our work makes use of the TCP/IP communications, based on the socket operation for the agents message interchange.
c) Agent Management System – this is a system that manages the access and the use of the agent platform; our work is based on agentMom, which is a framework upon which the distributed multiagent systems can be developed. It is implemented according to MaSE technology in Java and provides the basic building blocks for building agents, conversations between agents and the messages that are passed through conversations.

The MaSE – **M**ulti**a**gent **S**ystems **E**ngineering is a system that describes the rules between the agents, used to initiate and maintain the communications from agents structure. According to *Figure 1*, an agent starts one of its conversations as a Java thread. The communication establishes a socket with other conversation handler and sends the initial message. When the handler receives a message, it passes the message to the receive message method that validates the conversation. If the conversation is valid the agent starts its appropriate conversation as a separate Java thread. After this, the conversation thread from each agent controls all communication.

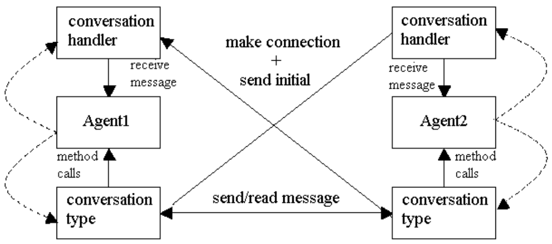
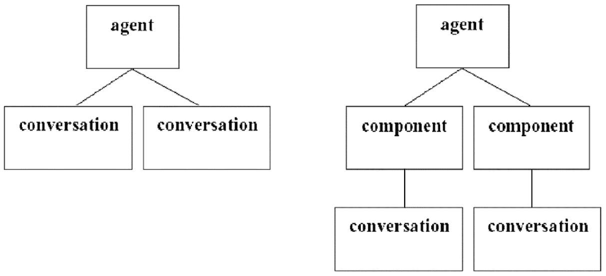


Figure 1. Conversations in MaSE technology based agentMom framework

According to [5], the agentMom framework is capable of using five types of conversations:
- Unicast conversations with TCP/IP; based on this conversation type there can be only two entities having communications at a time.
- Secured unicast conversations based on Secure Socket Layers (SSL) over TCP/IP; this type is the encrypted version of the above unicast conversations.
- Multicast conversations based on multicast socket and datagram packet. With this conversation type, an agent is capable of sending messages at a time to a group of agents that subscribes to the same multicast group.
- Secured multicast conversations based on multicast socket and datagram packet with symmetric key encryption algorithm, which represent the encrypted version of the above multicast conversations.
- Broadcast conversations including datagram socket and datagram packet, which allow the messages to be passed from one agent to all other agents in the organization at a time.

In addition to this, each agent is characterized by two structural types, which are illustrated in *Figure 2*. As one can see, an agent can be implemented following one of the two different structural types:
- Agent based structural type, where the agent must initialize and stop the conversations.
- Component based structural type, where the agent delegates the inner components to initializes and stops the conversations.



Agent-based conversations
Component-based conversations

Figure 2. Two different structural types of agents

According to the above classifications, in our work we implemented the unicast conversation type and component based structural type.

II. SYSTEM IMPLEMENTATION

Our simulation system contains three agents with specific function. We implemented this minimal structure in order to test our three agents based system.

Two of the agents are identical and they implement each of them a radio cell, A and B respectively, having a normalized radius of 1 and two sets of users. First set corresponds to fixed located users and has a number of 12 fixed located users.

Each user can make a call according to a log-normal probability distribution, with a mean value of λ=12 *calls/hour*, each call having a holding mean time of *ht*=120 seconds. We used a Dynamic Channel Allocation Scheme [6] in order to allocate a free channel to the new call from a total number of *ch*=35 radio channels. The system analyzes the CINR with 20 neighboring cells and if CINR >= CINR_{th} the channel is eligible and is allocated to the new call.

The second set corresponds to mobile located users. We considered a random number of users, which are mobile located, in the sense that they can change their location during call period and also they can cross the boundary between adjacent cells. The mobiles, *m*, are characterized by a selectable speed of velocity, *v*, and we tested them at $v = 5 \cdot \frac{R}{3600}$, where *R* is the normalized cell radius. Each mobile user can make a call according to a log-normal probability distribution, with a mean value of λ=12 *calls/hour* and the system allocates a channel according to a Dynamic Channel Allocation Scheme, from the total channels number of 35, which are common to fixed located users. If the mobile can use a free channel, then the number of mobile users is incremented and the user is moved toward the cell boundary. The starting point from each mobile is at the distance of 9/10 from the cell center and in the range between the starting point and the boundary, the system analyses the CINR for the corresponding mobile. If the CINR ranges between receiver threshold $r_{th}=5dB$ and transfer threshold $trans_{th}=10dB$, then the cell signalizes a transfer need. This transfer is then processed by the third agent, which deliberates if there is a free channel in the adjacent cell, in order to transfer the call to the cell B. If the mobile reaches the boundary without having a free channel in the adjacent cell, then a variable *forced calls* is incremented. The variable *total number of transfer needs* corresponds to the number of mobile users that have reached the cell boundary. The situation is illustrated in *Figure 3*. P_{bloc} is the blocking probability of the calls for the fixed located users, and is provided by (2):

$$P_{bloc} = \frac{\text{blocked calls}}{\text{total calls}} \tag{2}$$

P_{bloc_hand} is the forced calls probability for the mobile located calls, which need handover and is provided by (3):

$$P_{bloc\_hand} = \frac{\text{forced calls}}{\text{total number of transfer needs}} \tag{3}$$

The cell A is assigned to Agent1, cell B is assigned to Agent2 and Agent3 manages the handover between the two cells. After a mobile is transferred to the neighboring cell, it continues its movement until the opposite cell boundary is reached. The same mechanism is considered to the cell B in the transferring process of the mobiles to the cell A.

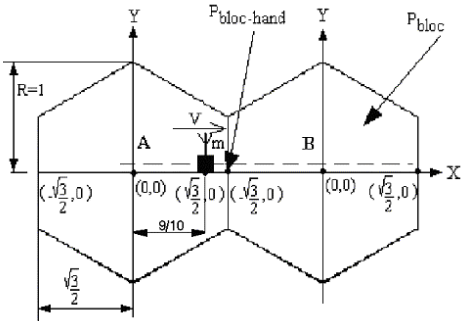


Figure 3. The two cells A and B considered in the experiment

Having these considerations, we implemented our three agents system following the specific steps in agentTool, defined by the Multiagent Systems Engineering:
- Capturing the goal Hierarchy
- Defining the Sequence Diagram
- Creating the Role Diagram
- Generating the Agent Template Diagram

The Goal Hierarchy step is illustrated in *Figure 4* and actually defines the general structure of the agents system goals. We defined the following goals architecture:
- The main goal of the hierarchy is to achieve the handover mechanism.
- The mid-level goals are represented by realizing the Base Station 1, Base Station 2 and Cost Function mechanisms.
- The lower level goals are BS1 simulation, BS1 transmit parameters, BS1 receive parameters, BS2 receive parameters, BS2 simulation, BS2 transmit parameters, CF receive parameters, CF transmit parameters, CF arbitration.

The BS1 simulation goal is to realize all the specific mechanisms in the base station from cell A, and BS2 simulation goal is to realize all the specific mechanisms in the base station from cell B.

The BS1 transmit parameters and BS2 transmit parameters goals are to send all specific parameters from Agent1 and Agent2 to Agent3. The CF receive parameters and CF transmit parameters goals are to send and to capture respectively all specific parameters from / to Agent3. The CF arbitration goal is to manage the allocation of the free channels to the handover needs. This mechanism consists in searching the channels in the corresponding cell and the first free channel found is assigned to the transferring call. After the Goal Hierarchy was defined, the next step was developed, namely the Sequences Diagram. Here we defined the sequences of the actions that the agents roles must follow. We illustrate the sequence resulted for our system in *Figure 5.*

In the horizontal dimension there are three roles defined in the entire system.

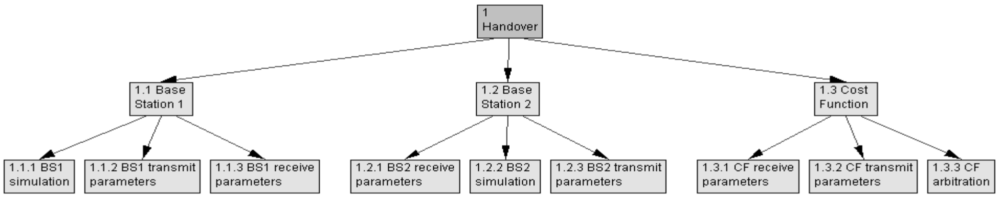


Figure 4. The goal Hierarchy of the implemented agents structure

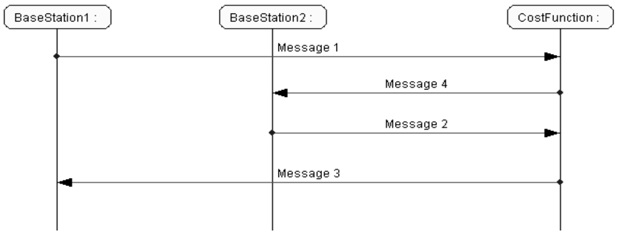


Figure 5. The Sequences Diagram

The agents from the system must carry out these roles.

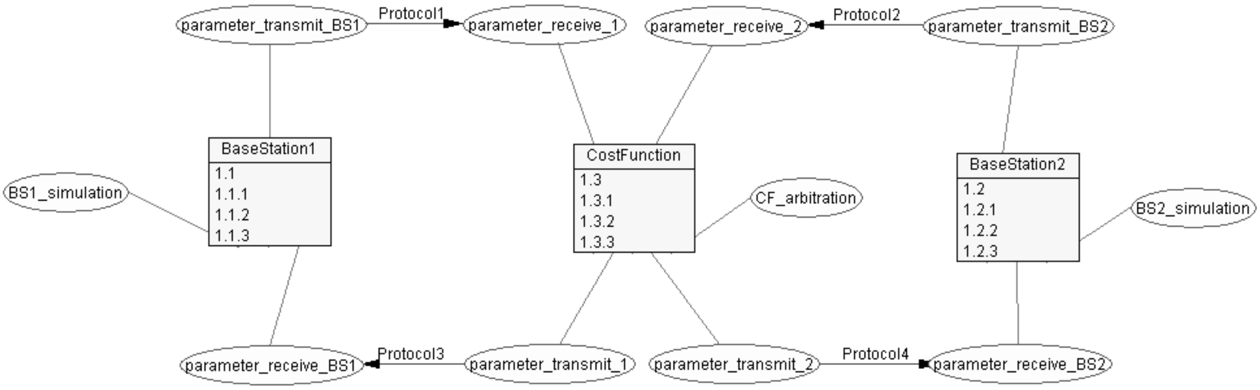
In order to do this, the roles interact each other by the mean of messages. These messages are passed through and this action denotes a sequence. In our system we defined 4 sequences with their respective 4 messages.

The next step is the Role Diagram. At this stage we developed the diagram illustrated in *Figure 6 (a).* Each of the 3 roles defined in the diagram communicates each other at the task level. A role can have more than one task and there can be distinguished 11 attached tasks, represented by oval shapes. Each task contains a set of routines that basically execute all of the specific actions of a component and thus, we implemented a component-based conversations system. The task is graphically defined and represents a finite automaton, including an initial and a final state and one or more intermediate states.

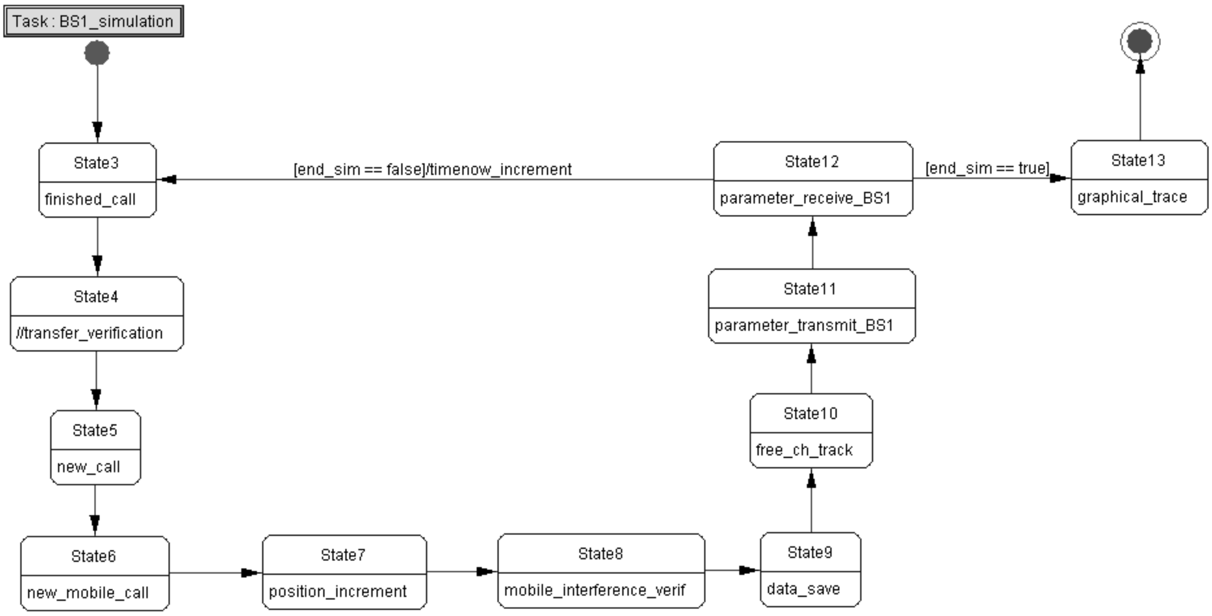
The transition between these fixed states can be made with or without any conditional selectors, and also with or without any associated actions. More details can be found in [7].

Following we present the structure of each task defined in our system.

- The BS1_simulation task has 13 states and is illustrated in *Figure 6 (b).*



a) The Role Diagram



b) The BS1_simulation task

Figure 6. a) The Role Diagram, b) The BS1_simulation task

- The parameter_transmit_BS1 task has 3 states and the stucture is similar to parameter_transmit_1, parameter_transmit_2, parameter_transmit_BS2 tasks. We present the structure is *Figure 7 a).*
- The parameter_receive_BS1 task has 3 states and is similar to parameter_receive_1, parameter_receive_2, parameter_receive_BS2 tasks. We present the structure in *Figure 7 b).*
- The CF_arbitration task has 8 states and is illustrated in *Figure 7 c).*
- The BS2_simulation task has 13 states and is illustrated in *Figure 7 d).*

The Agent Template is the final step of the system definition and the diagram is illustrated in *Figure 8 a).*

Each conversation is automatically generated by the agentTool platform and has two sections: the Initiator and the Responder, as illustrated in *Figure 8 b).* The Initiator section is located on the transmitting task, whilst the Responder section is located on the receiving task.

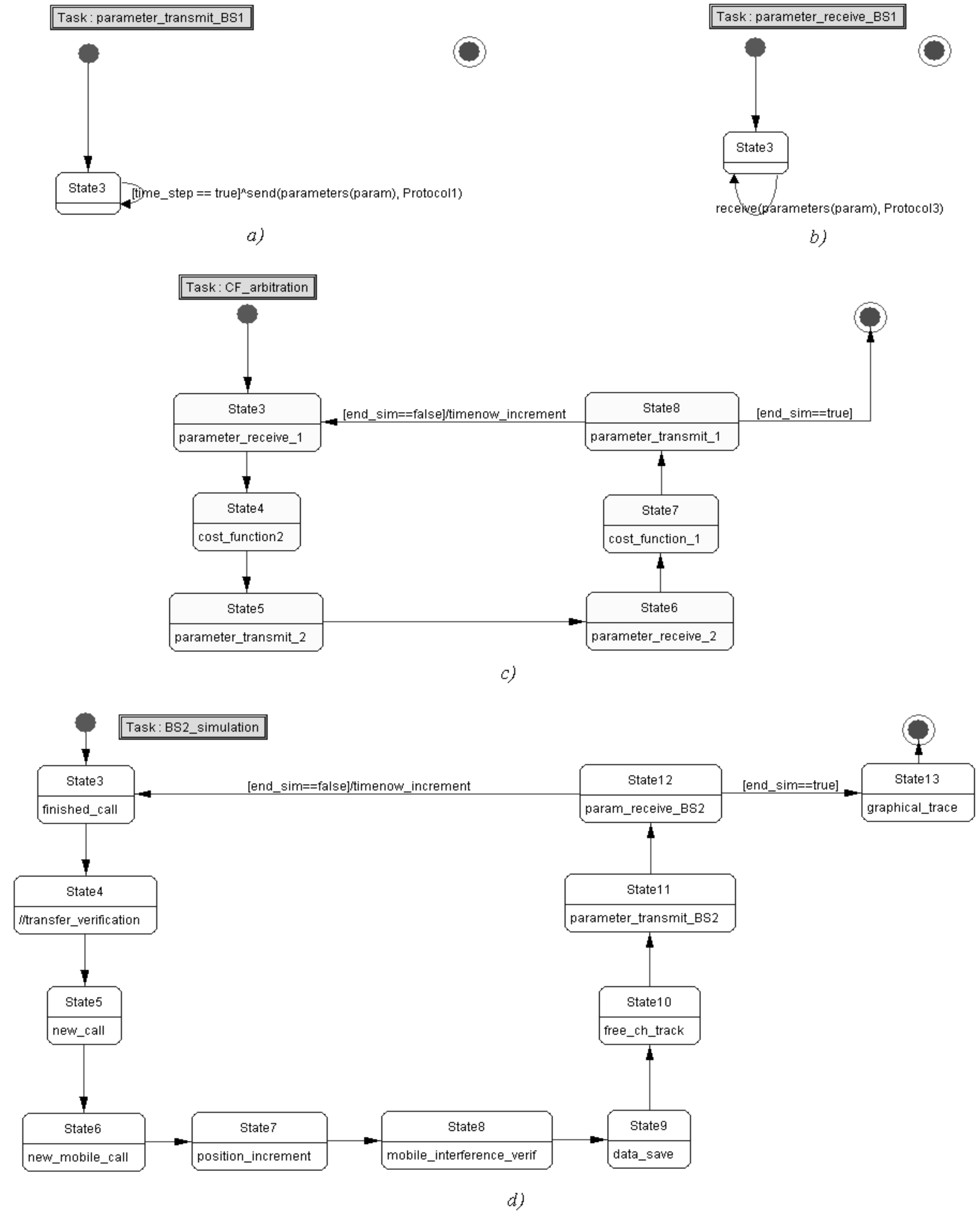


Figure 7. a) The parameter_transmit_BS1 task, b) The parameter_receive_BS1 task, c) The CF_arbitration_task, d) The BS2_simulation task

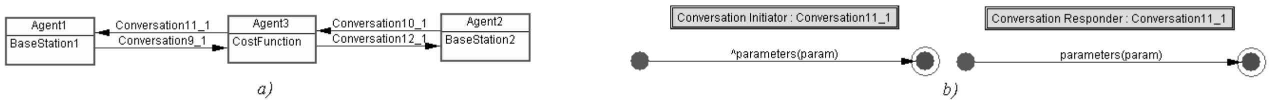


Figure 8. a) The Agent Template Diagram, b) The Initiator and the Responder Conversations

III. THE RESULTS

In our simulation we used three computers with Microsoft Windows operating system, having Java 1.6 environment installed. The computers were physically interconnected with Ethernet link, and logically addressed with TCP/IP protocol. On two computers we simulated two radio cells, cell A and cell B, with the aim of Agent1, respective Agent2, having 12 fixed positioned users and a random number of mobile positioned users, whilst on the third computer we used Agent3 in order to manage the mobile transfers between the two cells. The cost function implemented in Agent3 is the simplest one and we used it in order to verify our system: the first unallocated channel found in the neighboring cell from the amount of channels is allocated to the transferring mobile. If the resulting CINR is greater then a threshold value of 18*dB*, the channel is accepted to be used for the corresponding transfer. Because we used the same conditions for fixed located users as for mobile located users concerning the call generation mean value of 12 *calls/hour* and the threshold CINR of 18 *dB*, and also we used the same channels set for both fixed and mobile located users categories, we expected to obtain the same values for the blocking calls probability, as for the forced calls probability, which characterizes the fixed located users and the mobile located users, respective. Our system used a simulated time of 20000 seconds with an increment of 10 seconds in every simulated cycle. Agent1 and Agent2 respectively, simulated a number of 21 cells with a total number of 35 channels for both fixed and mobile users and the results are provided in *Figure 9* for Agent1 and *Figure 10* for Agent2.

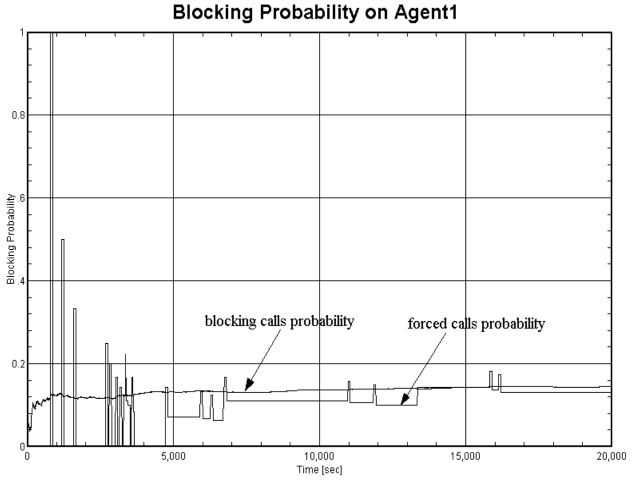


Figure 9. Blocked calls probability versus forced calls probability in Agent 1

As one can see, the mean value of 0.14 for blocking probability is approximately the same for the forced calls probability value, for both Agent1 and Agent2. We used three Windows based computers, each at 1.8 *GHz*. For 20000 simulated *seconds*, our system needed a real time of 80 *minutes*, or approximately 4800 *seconds*.

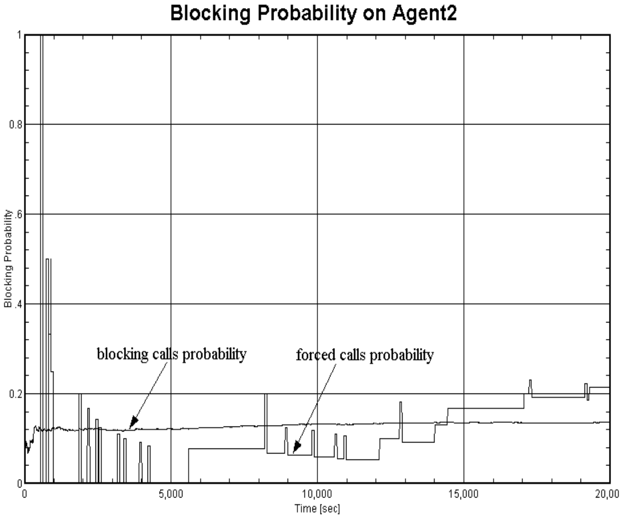


Figure 10. Blocked calls probability versus forced calls probability in Agent2

IV. CONCLUSSIONS AND FUTURE WORK

We designed a minimal structure containing three agents, in order to simulate complex situations for handover analyze. The agents implemented the Dynamic Channel Allocation Schemes for channel management and the final goal of our system was to arbitrate the mobile transfers between two adjacent cells. Our minimal structure can be further multiplied in order to obtain a closer behavior to the real conditions. In our work we focused on Multiagent Systems Engineering technology, used by the agentTool Platform. The resulted code for the agents structure was automatically generated in Java, which is operating system independent. The tests for our minimal structure revealed a correct behavior for the imposed conditions, reflected in graphical results.

REFERENCES

[1] S. H. Oh et al., “Prioritized Channel Assignment in a Cellular Radio Network”, *IEEE Transactions on Communications*, vol. 40, pp. 1259-1269.

[2] H.Harada, R. Prasad, “Simulations and Software Radio for Mobile Communications”, *Artech House*, London, 2002.

[3] Scott A. O’Malley, Athie L. Self and Scott A. Deloach, “Comparing Performance of Static Versus Mobile Multiagent Systems”, *National Aerospace and Electronics Conference (NAECON)* Dayton, OH, October 10-12, 2000.

[4] ”AgentTool 1.8.3 User’s Mannual” *AF Institute of Technology*, June 2001

[5] Chairoj Mekprasertvit, “Applying Broadcasting/ Multicasting/ Secured Communication to agentMom in Multi-Agent Systems – User’s Mannual”, *Kansas State University*, Spring 2004.

[6] Gabriel Sirbu and Ion Bogdan “Performances Analysis of Different Channel Allocation Schemes for Personal Mobile Communication Networks”, *Proceedings of the WSEAS International Conference of Dynamical Systems and Control*, Venice, Italy, 2005.

[7] A. S. DeLoach “Analysis and Design Using MaSE and agentTool”, *Proceedings of the 12th Midwest Artificial Intelligence and Cognitive Science Conference (MAICS)*, 2001, http://www.macr.cis.ksu.edu/projects/agentTool.