Intelligent features within the J-Sim simulation environment

Nada Meskaoui[1,3], Dominique Gaiti[1,2], Karim Kabalan[3]

[1] LIP6-Université de Paris 6, 8 rue du Capitaine Scott
75015 Paris, France
[2] LM2S: Université de Technologie de Troyes, 12 rue Marie Curie
10010 Troyes Cedex, France
[3] American University of Beirut,
Faculty of Engineering and Architecture, Lebanon

Abstract. This paper presents our approach integrating a high level of intelligence within telecommunication networks. Artificial intelligence has proved its efficiency in resolving complex issues in different fields like air traffic control, robotics, and medical diagnosis. However its usage in telecommunication networks is still very moderated. The main objectives of our framework is to encourage the deployment of techniques characterized by a high level of intelligence – inspired from the artificial intelligence domain – in networking. For this reason, we build an extension, with intelligent capabilities, to the J-Sim simulation environment. This extension proposes an intelligent package having the essential entities required for simulating intelligent features. Using this intelligent package and the abstract node model (INET) proposed by J-Sim for network simulations, it will be possible to test the efficiency of integrating intelligent techniques in networking.

1 Introduction

The last few years have perceived great increase in the research efforts in different domains. In telecommunication, for example, the exponential evolution in the usage of the Internet and mobile networks along with the demands for better quality of treatments within the transit networks have encouraged the researchers to think about and propose new techniques and networking solutions to improve the network performance. Also, in the field of artificial intelligence, the needs for human like, dynamic and intelligent solutions has led to enforce the efforts in studying the efficiency of integrating intelligent features based on intelligent agents and multi-agent systems in different domains like the study of the consumer behavior in a competitive market or the assistance of the drivers in finding the best road to destination and data management and retrieval.

In networking, agents - especially mobile agents - were introduced to collect information, to discover the network topology and control the global functioning of the network. Therefore, solutions introducing a high level of reasoning in network manage-

ment and control are not yet available. This paper proposes a framework for testing high level of intelligence within telecommunication networks, based on the J-Sim simulation environment [1].

2 J-Sim

J-Sim [1] is an open, component-based, compositional simulation environment, built entirely in Java upon the notion of the autonomous component architecture. It is a truly platform-neutral, extensible, and reusable environment as a system is composed of individual components able to be plugged into a software system, even during execution.

J-Sim proposes an implementation of an abstract network model (INET) on the top of the component-based architecture. This model defines the generic structure of a node and several generic network components with their associated contracts. INET proposes an internal node structure of the hosts and routers in two layers, which are the Upper Protocol Layer (UPL) that contains transport, routing, signaling, and application protocol modules and the Core Service Layer (CSL) that provides a set of well-defined services, which are common in most network architectures, to modules in the UPL.

The J-Sim working group has also proposed an implementation of some techniques like DiffServ and Intserv and recently an implementation of both MPLS and active networks were proposed.

3 Intelligent features

Traditionally, intelligent features are inspired from the artificial intelligence domain and based on the concepts of intelligent agents and multi-agent systems. Agents and multi-agents systems (MAS) are two innovative and interesting concepts for a great number of researchers in different domains like road traffic control, biologic and social phenomena simulations, industrial applications, and others.

Agents are characterized by their autonomy, their ability to interact with other peers and devices. They can learn, plan future tasks and are able to react and to change their behavior according to the changes in their environment.

A multi-agent system is a group of agents able to interact and cooperate in order to reach a specific objective.

3.1 Agents

The concept of agents has occupied great importance in the domain of Artificial Intelligence (AI) where different interpretations and definitions were associated to the word "Agent". [2] considers that agents are intelligent entities that act according to their knowledge. [3] defines agents as high-level autonomous entities able to act in an

environment where other agents and devices execute and [4] considers an agent as a person managing an agency or something that initiates a chemical reaction or an organism that causes a disease. Finally [6] considers that all these definitions confirm that the question "what is an agent?" is as embarrassing as the question "what is the intelligence?". It is so better not to confine to one definition and to define an agent according to its field of application.

Agents are characterized by their properties that determine their capabilities. Different properties are defined – like autonomy, proactive-ness, flexibility, adaptability, ability to collaborate and coordinate tasks and mobility. The agent according to its role within its environment acquires one or some of these properties.

Different types of agents can be defined like reactive, cognitive, mobile, hybrid, collaborative and others. Each of these types is characterized by some properties like for example autonomy, mobility, reactivity, pro-activeness, and collaboration.

3.2   Multi-agent systems

A multi-agent system is a group of agents defined as "a loosely-coupled network of problem solvers that work together to solve that problems that are beyond their individual capabilities" [7]. These problem solvers are agents and are defined as autonomous entities able of acting in their environment. They are able to communicate directly with other agents, posses their own resources, perceive their environment (but to a limited extent), and have a behavior [5]. In the frame of this communication, agents cooperate in order to insure coherent behavior within the MAS. This cooperation, according to [8], lets the agent know the objectives of a peer and therefore adopt these objectives as its proper objectives in order to improve the global behavior of the MAS.

3.3.   Agents in telecommunication networks

In the last few years, the research efforts was increased in the domain of telecommunication to study the ability of new techniques based on agents – especially mobile agents – and multi-agent systems in designing networks and network applications;

[9] presents a mobile-agents approach, highly distributed and decentralized with agents spread across the network working to accumulate connectivity information. The agents in this system move around the network, discover topology information, and carry with them the information they have gathered as they explore. Results prove that the cooperation between the agents and the diversity of behaviors between collaborating agents greatly improve the performance of the whole system.

Other researchers like [10] and [11], have borrowed metaphors from biological systems to develop higher-level network management frameworks. These frameworks rely on the ability of social insects to solve problems, sometimes-difficult problems, in a distributed way, without a central control, on the basis of local information. [10] and [11] used these abilities for network control in a telecommunication network,

specifically to get dynamic update of the routing tables of the network nodes according to the different nodes' loads.

3.4 Platforms simulating intelligent features

Agents and MAS oriented programming is an advanced software-modeling paradigm proposed by researches in the distributed artificial intelligence domain. It addresses the need for software systems to exhibit rational, human-like behavior in different domains.

MAS platforms are characterized by their architectures able to build systems that work on the full range of tasks expected from an intelligent agent, from highly routine to extremely difficult, open-ended problems. They have the ability to represent and use appropriate forms of knowledge and a full range of problem solving methods. These types of platforms are goal-oriented, based on states and operators. They continually try to select and apply operators to states until the achievement of a specific goal.

Different platforms were proposed for multi-agent systems simulation like JAFMAS, DECAF, JACK, DEMAS and others, each having its proper implementation and fields of application.

4   The proposed model

To introduce the concept of agents and multi-agent systems within the J-Sim node model, we propose to extend it with an intelligent package. This package represents the intelligent aspect of the whole system and implements the various classes that may serve as the building blocks for specifying and communicating agents. This means that all the entities required in a standard environment for multi-agent systems simulation should be implemented. These entities are: the communication between agents, the agent's resources, data base and rule base, the social model, and the linguistic features, as shows fig. 1.
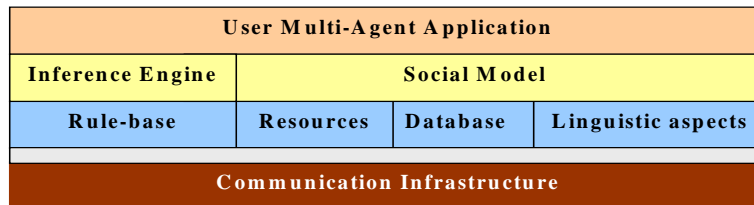
User Multi-Agent Application			
Inference Engine	Social Model		
Rule-base	Resources	Database	Linguistic aspects
Communication Infrastructure			

Fig. 1. Organization of the intelligent package

The user, according to its agents' type and behavior, defines the multi-agent system application. This application benefits from the services provided by the social model, which determines the way the agent communicates, cooperates and interacts with

other agents of the system in order to bring about a coherent solution. During its communication, the agent exchanges messages using a common agent-independent language, defined by the Linguistic layer. In these messages, the agent exchanges resources and knowledge and then stores acquired data in the data model. To reason on this data, the agent implements a rule-base that contains the needed rules to decide about its future actions and invokes these rules using the inference engine.

Finally, we consider that the communication infrastructure is provided by the standard components proposed by J-Sim to connect and communicate different entities.

5   The intelligent package - java classes

We propose an implementation of the agent package in 18 main java classes that provide the essential elements and entities for the developers of intelligent applications within a telecommunication network. Five of these classes, related to the communication aspects, were originally taken from the multi-agent system platform JAFMAS, with some modifications to adapt them to our proposed model. The defined classes and the relationship between them are described in Fig. 2. It shows how the classes use each other's methods in order to build a coordinated framework.

The "Agent" class is implemented as an abstract component within J-Sim. This class provides the standard attributes and methods to be used by any agent type related to any possible application. It should be extended to define the agent related to a specific application. This agent requires interacting with different java classes in order to cooperate with other peers, to exchange messages and store data.

The social model is implemented in a list of java classes. The main are "Conversation" and "ConvRule". The linguistic aspect is mainly provided by the "IMessage" class that describes the format of the messages exchanged between the agents.

After the reception of a message, the agent treats it or directs it to its concerned conversation in case the message is sent in the context of a conversation between the agents. The "MsgRouter" java class is responsible of the routing of the received messages to their relative conversations. When the conversation receives a message, it stores it in its messages' queue using the "MsgQueue" class.

We consider that the agent may require communicating with other agents within the multi-agent system according to the Contract-Net protocol. For this reason, an implementation of this protocol was proposed in both the "ContractNet" and "Contract-NetReply" classes.

The three java classes – "BeliefSet", "Belief_Sets" and "data" – are defined to manage and store data. An instance of the "Belief_Sets" class stores the different instances of a specific BeliefSet and provides methods to add, remove and query these instances and the data class defines the structure of an element of the BeliefSet.
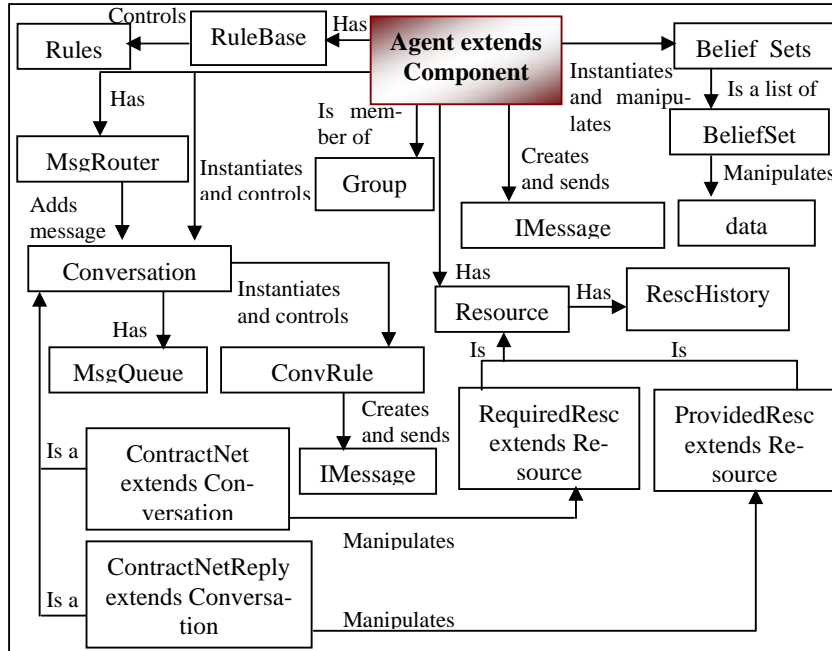
Fig. 2. The main classes' relationship diagram

The "Resources" class is an abstract java class that defines the common elements to the different types of resources. The "RequiredResc" and "ProvidedResc" classes that define respectively the elements and methods related to the required and provided resources extend this class and the user, to implement some methods according to the application requirements, should also extend the "RequiredResc" and "ProvidedResc" classes. The third class related to resources is the "RescHistory" class conceived to store the history of a resource.

The information concerning the different groups defined within the multi-agent system is stored in the "Group" class. This class extends the "Belief_Set" java class. Finally, the classes related to the representation and management of the production rules are "Rules" and "RuleBase".

6. Case Study

In this section we propose a case study integrating a multi-agent system within a transit network that provides different levels of treatments for different types of traffics, in the objective of preventing congestions.

In the following simulations we consider that two sources S1 and S2 send respectively to the destinations D1 and D2 traffics with two different levels of quality of

services, where the traffic sent by S1 demands a higher level of quality of service than the traffic sent by S2. As both traffics transit the network (see Fig. 3), the S1 traffic is treated in priority to the S2 traffic in case of congestion. The implemented agent behavior tries to increase the S2 traffic throughput and to balance the traffic load within the network, if congestions occur. The simulations' results show great increase in the throughput of the S2 traffic (see Fig. 5) if agents are integrated within the transit network.
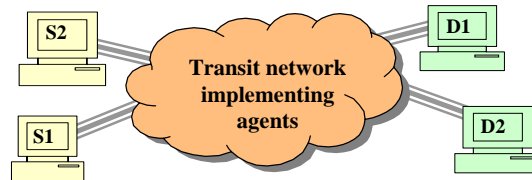

Figure 3. *Global view of the network*

In case agents are not integrated within the network, great loss of the S2 traffic is perceived as shows Fig. 4. Finally, Fig.   1 traffic modifies its default road to destination due to a congested situation   ork's links. This behavior relieves the congestion and permits better use of the deployed resources within the network.
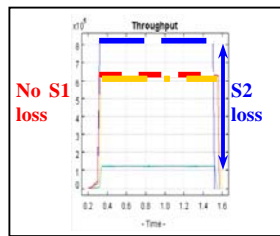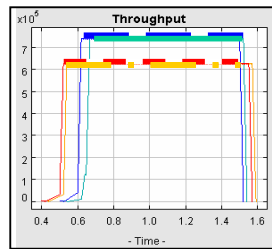

Figure 4. *Transit network without agent's integration*

Figure 5. *The throughput of the S1 and S2 traffic after the agent's integration*
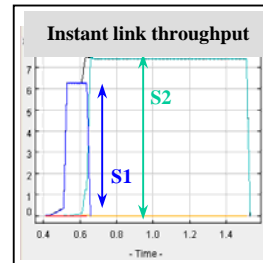
Figure 6. *The throughput of the congested link. S1 is redirected to resolve the congestion*

7. Conclusion

This paper presents our approach that builds an extension to the J-Sim simulation environment with intelligent capabilities. This extension provides an infrastructure for the validation of different techniques based on a high level of intelligence, in networking. Some applications integrating agents within different types of networks were tested using this extension [12], [13]. Simulations show great increase in the network performance after the integration of intelligent features. This means that our proposal could be considered as a first step towards both filling the gap between the

artificial intelligence domain and networking and investigating intelligent capabilities in the world of telecommunications.

8. References

1. J-Sim team, Ohio State University, available at: www.J-Sim.org, updated 2004.
2. Demazeau Y. and Müller J.P.: Decentralized AI2. Editions North-Holland Elsevier Science B. V, 1991.
3. Shoham Y.: Agent Oriented Programming. Artificial Intelligence, 1993.
4. Minsky M., A conversation with Marvin Minsky, Communication of the ACM, 1994.
5. Ferber J.: Multi-Agent System: An Introduction to Distributed Artificial Intelligence. Harlow: Addison Wesley Longman, 1999.
6. Guessoum Z.: Un environnement opérationnel de conception et de réalisation de systèmes multi-agents. P.H.D. report, University of Paris VI, 1996.
7. Durfee E. H., Lesser V. and Corkill D. D.: Trends in cooperative distributed problem solving. IEEE transaction on knowledge and Data Engineering, vol. KDE-1, pp. 63-83, Mar 1989.
8. Gallier J.R.: Atheorical Framework for Computer Models of Cooperative Dialogue, Acknowledging Multi-Agent Conflict. P.H.D. report, Open University (UK), 1988.
9. Minar N., Kramer K and Maes P., "Cooperating Mobile Agents for Mapping Networks". In Proceedings of the First Hungarian National Conference on Agent Based Computing, May 24, 1998.
10. Schoonderwoerd R., Holland O., Bruten J. and Rothkrantz. L., "Ant-based Load Balancing in Telecommunications Networks. Adaptive Behavior", 5(2):169-207, available at: http://www-uk.hpl.hp.com/people/ruud/abc.html, 1997.
11. Bonabeau E., Henaux F., Guérin S., Snyers D., Kuntz P., and Theraulaz G., "Routing in Telecommunications Networks with "Smart" Ant-Like Agents". In Intelligent Agents for Telecommunications Applications'98, availableat: http://www.santafe.edu/sfi/publications/Abstracts/98-01-003abs.html, 1998.
12. N. Meskaoui, D. Gaiti, K. Kabalan: Implementation of a multi-agent system within a Diffserv network to improve its performance. In Proc. ISE2003, Montreal, Canada July 2003.
13. N. Meskaoui,: A framework to model and simulate multi-agent systems within telecommunication networks: new environment, tools and behaviors. P.H.D. report, Paris 6 University, 2004. France.