



Asset Management System through the design of a Jadex Agent System

Javier Carbo^a, Jose M. Molina^a and Miguel A. Patricio^a

^a Computer Science Department, University Carlos III of Madrid, Av: Universidad 30, Leganés 28911 Madrid
jcarbo@inf.uc3m.es, molina@ia.uc3m.es, mpatrici@inf.uc3m.es

KEYWORD

Agents; Asset Management; Recommender Systems; Planning;

ABSTRACT

In this paper we have designed an agent system using JADEX platform in order to facilitate the asset management of an institution with several types of assets with some unknown similarities, and a high number of operators. Such environment would justify the use of a recommendation system that would provide an useful plan to be applied over an asset generated from previous maintenance operations hold over other assets. Here we provide the full design of such agent system, including the ontology, beliefs, plans, protocols and goals. This design has been implemented within GAIA: an information & asset management solution developed by Altran.

1. Introduction

The aim of an asset management system is to maintain a set of assets in a condition that meets the requirements and preferences of a community, so as to prevent any service interruption. Despite the importance of infrastructure in economic and social development, resources available for conservation are often inadequate, so an accelerated deterioration thereof is generated (Schneider et al, 2006). In this context, asset management discipline appears, which seeks to optimize the allocation of resources for the management, operation and maintenance of infrastructure through an analysis of their life cycle.

Any organization whose service has involved a set of assets with high performance and availability of their facilities must have a strong ability to deliver and provide the best value. A clear example is the municipal or local agencies, which face challenges of high level, as budgetary constraints, legal regulations and new demands for services. Many local governments (cities, towns and metropolitan areas) should work with reductions in state contributions, or tax revenues, while, on the other hand, are forced to expand their services despite not having adequate infrastructure. In this environment, where should take more actions with fewer resources, achieving greater operational efficiency through improved asset and services management is essential.

In this context, asset management systems play a major role. These systems are the basis of the activity of the utility, which is developed based on the operation and maintenance of a very large number of assets and facilities that are widely dispersed geographically. To date, these systems have been developed in a complementary manner and following the paradigm of Geographic Information Systems (Zhang et al, 2009). These systems suffer from a number of limitations that make their use not effective from the functional point of view. Through these systems you can maintain a centralized model of assets, always oriented to their geographical organization. This is achieved by building layers containing the "homogeneous" information, for instance:

- Bins Layer: Layer where the bins of the city are represented
- Layer semaphores: Layer where the lights of the city are represented

- Layer neighborhoods: Where the boundaries that define the neighborhoods are saved.
- Layer parks: Where city parks are delimited.

Figure 1 shows this method of administration:

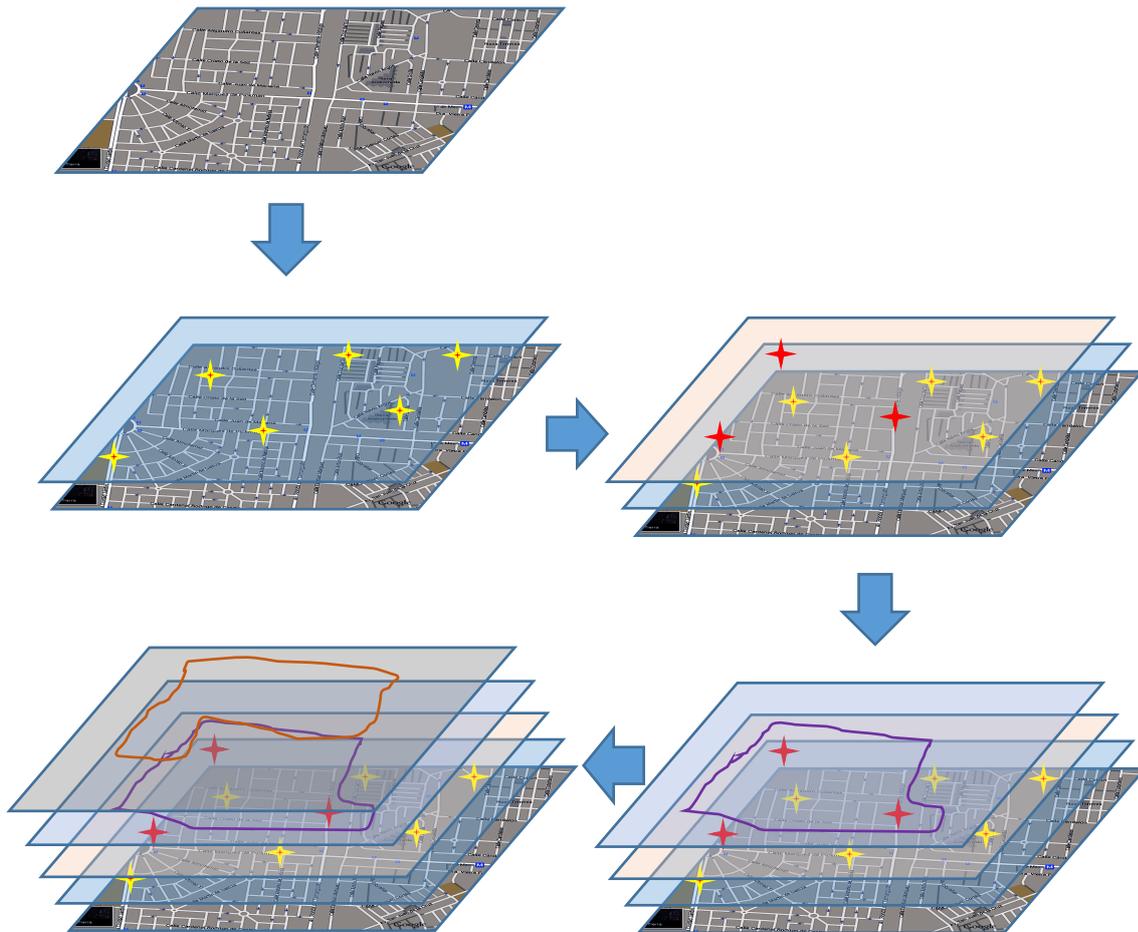


Figure 1. Example of GIS-oriented asset management system.

However, GIS-oriented asset management systems presents serious drawbacks which can be summarized as follows:

- [1] It requires an enormous amount of hardware resources that support all elements required in the different layers.
- [2] It has a high cost of licenses if using proprietary software
- [3] It is only possible to make information crossings by geographical areas, not administrative or otherwise.

- [4] Requires knowledge of GIS by users.
- [5] Requires technical expertise in GIS development and system administration.
- [6] Perform complex analysis of several layers can be very expensive.
- [7] It is virtually impossible management elements found in warehouses or buildings.
- [8] The management of mobile elements such as emergency services, waste collection services, etc., is complex

Thus, new asset management systems are emerging, like GAIA, developed by Altran and used as the basis to test the design in a real environment, and are oriented to a functional organization of assets. This new paradigm will allow significant optimization in terms of speed and access, as well as more technological novelty, allow a functional asset management. This management can treat the assets according to their functional utility, and its organization is based on graph theory. This new model represents a substantial technological innovation over existing systems and is a new way of organizing information within an asset management system. In this new paradigm the information is organized in directed graphs.

This new approach is an important optimization, however, the structure of directed graphs can reach great complexity. This complexity may be greater, if we store the history of actions taken during the life of the asset management. This complexity could be a drawback by the staff in charge of its management. For this reason, it has been introduced in this paper a new algorithm in the area of Artificial Intelligence, called "Cooperation on intelligent agent systems based on experience". By the intelligent agent theory, the authors develop a new algorithmic that allows to manage and capture the experience of more experienced staff in order to support other users with less experience in the asset management system. Therefore, it will be a distributed system consisting of multiple sources of information (generated by the experience of each of the users who have previously interacted with the tool). The novice user will have to merge previous experience of these users, so it will be necessary to introduce the concept of trust and reputation of the information of the user.

2. Agent Systems and Asset Management Systems

Agents have been used largely to solve many type of problems of distributed nature (Nwana, 1996). This technology provides a high level of encapsulation and parallelism useful in coordination and competition domains. It has also been largely applied in planning (Ephrati & Rosenschein 1993; Georgeff, 1983) and recommending problems (cita recommending). Among all the approaches to Agents, deliberative paradigm has the most human-like reasoning process. It is often denoted as an intentional system (Dennet, 1987), since decisions hold on the future actions-to-be-taken take the form of intentions. The most extended intentional approach is the so called BDI paradigm (Rao and Georgeff, 1995), where internal reasoning of agents is computed using a combination of three elements: beliefs, desires and intentions. These elements are high-level abstractions of commonly-used concepts that humans apply in daily life decision taking. In summary, these elements are abstraction tools that provide an useful way to describe, analyze and predict the behaviour of complex systems (which are the intended application domain of agents).

Beliefs are statements about itself and the outside world that agents take as truthful although they just perceptions that could be uncertain or incomplete. They describe the "informational state" of agents. Desires are the description of future preferred situations (particular setups of the outside world). Desires express the motivational state of an agent. Although in some sources, desires are confused with goals, goals include an active commitment of the agent, and a higher level of concretion than desires. At last, intentions are actions chosen to be executed by the agent in the near future. They capture the deliberative state of the agent.

While reactive (non-deliberative) agent systems claim to provide close to real-time efficiency, BDI agents often face with higher success the requirements of complex domains where dynamicity, uncertainty, limited observability and

resources are involved. This is the case because beliefs capture the essence of the changes that are taking place, and since perception is incomplete, beliefs just represent what is (and is not) actually perceived. Furthermore, beliefs should synthesize just the relevant information of all that was perceived by agent sensors/communications. Desires (and Goals) explicitly show the motivation of agents, the reason why such agents are running, it allows fault recoveries and take the opportunities when they arise. Intentions allow agents to adapt their actions to a changing world where urgency of actions-to-be-taken can change, while long-term reasoning is also addressed.

We have applied such BDI approach using JADE platform (Braubach et al., 2004), JADE was developed in JAVA language by Hamburg University in order to extend the functionalities of the most used agent platform: JADE (Bellifemine et al., 2007). Such extension is focused on implementing BDI model. JADEX distinguishes between the explicit motivational state described with goals in an XML file, while the way they are achieved is represented by plans in JAVA. By this distinction, JADEX allows the inclusion of the corresponding activation conditions of goals and their relationship with JAVA plans and JAVA beliefs for each type of agent. In this XML file are also defined two kinds of events: external (caused by the reception of a message), and internal (caused by changes applied by own plans and beliefs). JADEX allows four types of goals: perform an action, achieve a given situation, query information, and maintain a situation. JADEX itself invokes the JAVA execution of plans corresponding to the adopted goals when activation conditions met. Inside this plan, java beliefs can be accessed and modified accordingly to the plan definition, and also read the content of received messages, or send new messages to other agents.

Such JADEX agents act in a particular context where a big institution has to apply maintenance operations over a high number of assets, which are categorized according to a computable similarity. Such institution also has access (communication) to a given number of operators that execute such maintenance operations in a particular sequence forming a plan, where we do not know which interdependence such actions have among them (or if such relationship even exists). So the operations that compose a maintenance plan over an asset has been decided according to the recommendations from previous success/failures of maintenance plans executed by several operators. Such success/failure level is estimated from the time frequency of assets requiring additional maintenance operations.

Also, not all the operators have the same ability to implement such plans, and since some of them have more expertise than others, such factor has also to be taken into account in order to compose the final maintenance operations plan. This ability is just estimated in such a way similar to the reputation of agents in models of trust applied to Agent systems (Josang et al., 2006; Sabater-Mir and Sierra, 2005).

3. FIPA-compliant Ontology

A JADEX ontology was defined following the recommendations of FIPA standard (Poslad & Charlton, 2001). Our ontology is composed by three types of elements: concepts, predicates and actions. Actions describe the task to be performed over a set of concepts, while predicates are statement over the properties of a set of concepts. Finally, concepts are just an structured information descriptive elements. In our domain of maintenance operations of assets, we distinguish the next concepts:

- NodeID: Agent identifier of an agent acting as an asset
- NodeType: Type and subtype of the asset represented by the agent. This means that assets are classified according to a 2-level categories.
- Operation: Maintenance operation to be taken over an asset. This operation is atomic, and a sequence of them would compose a maintenance plan. An operation is defined by its own type (a numeric value), and the complete type of the element over which it was applied.
- Expertise: is a measure of the ability showed by an operator executing operation plans. It has two properties: global expertise, general ability that is achieved by practice applying any kind of operation; and local expertise, specific ability shown over a particular type of asset, computed from the success achieved in the

right composition of operation plans. These two values will determine how much time the asset would take before requiring another maintenance operation.

The predicates of our infrastructure maintenance domain are the next ones:

- hasExpertise: it states whether a particular operator has or not expertise with asset of a given type. It is related with concept NodeType.
- ownExpertise: it states which particular expertise an operator has over assets of a given type. It is related with Expertise and NodeType.
- isOperator: it states whether the agent knows about operators with expertise in assets of a given type. It is related with NodeType.
- knownOperators: it states which known operators have expertise in assets of a given type. It is related with Agent Identifier and NoteType.
- recommends: It states whether an operator recommends an operation plan to be taken over a given type of infrastructure node. It is related with the concept NodeType.
- recommended: It states the operation plan that an operator recommends to be taken over a given type of asset. It is related with NodeType and (list-of) Operation.

The actions of our infrastructure domain are the next ones:

- assign: Action of assigning a particular asset of a given type to an operator agent. It is related with the concepts: NodeID and NoteType.
- update: Action of updating (application of a maintenance operation plan) a particular asset of a given type. It is related with the concepts NodeID and NodeType.
- execute: Action of executing a particular sequence of operations over a given type of asset. It is related with the concepts NodeType and Operation.

4. Agent Communications

JADEX agents communicate between them using ordered sequences of FIPA-compliant messages that compose predefined interaction protocols. The protocols we used in our agent system are slight variations of the query and request protocols that FIPA standard suggest.

Protocol 1: Assigning an operator to an asset. This protocol is composed by 3 messages between node and operator agents: a call for proposals (FIPA performative cfp), the corresponding responses in form of proposals (FIPA performative propose), and the final corresponding acceptance/rejection of such proposals (FIPA performatives accept/reject-proposal).

In advance we define the minimum elements that FIPA messages require: name of the message, communicative act or performative, sender and receiver(s), content (exchange information included in the message, expressed in ontology terms, java classes in our system).

| Message # | Performative | Sender | Receiver | Content | Linked concepts |
|-----------|---------------|-------------|-------------|----------------|------------------|
| 11 | Cfp | Node | Operator(s) | Action: assign | NodeID, NoteType |
| 12 | Propose | Operator(s) | Node | Action: assign | NodeID, NoteType |
| 13 | Accept/reject | Node | Operator(s) | Action: assign | NodeID, |

| | | | | | |
|--|-----------|--|--|--|----------|
| | proposals | | | | NoteType |
|--|-----------|--|--|--|----------|

Protocol 2: Proposing maintenance operation plan. This protocol is composed by 2 messages between node and operator agents: a proposal of operations and the corresponding acceptance of such proposal.

| Message # | Performative | Sender | Receiver | Content | Linked concepts |
|-----------|-----------------|----------|----------|----------------|------------------|
| 21 | Propose | Node | Operator | Action: update | NodeID, NoteType |
| 22 | Accept proposal | Operator | Node | Action: update | NodeID, NoteType |

Protocol 3: requesting the execution of a maintenance operations plan. This protocol is composed of 2 messages: a request of executing the operation list included in the content of the message, and the corresponding response informing about the new expertise that become modified by the node agent.

| Message # | Performative | Sender | Receiver | Content | Linked concepts |
|-----------|--------------|----------|----------|----------------------------|-----------------------------------------|
| 31 | Request | Operator | Node | Action: execute operations | Expertise, List of Operations, NoteType |
| 32 | Inform done | Node | Operator | Action: execute operations | Expertise, List of Operations, NoteType |

Protocol 4: querying about recommenders. This protocol is composed by 2 messages: querying an operator about the known operators who are specialized in the subtype of the desired asset, and the corresponding response with the list of AgentIdentifiers.

| Message # | Performative | Sender | Receiver | Content | Linked concepts |
|-----------|--------------|-------------|-------------|---------------------------|------------------------------------|
| 41 | Query | Operator | Operator(s) | Predicate: isOperator | NoteType |
| 42 | Inform ref | Operator(s) | Operator | Predicate: knownOperators | List of AgentIdentifiers, NoteType |

Protocol 5: querying about the expertise of recommenders. This protocol is composed by 2 messages: querying an operator about its expertise in the subtype of the desired asset, and the corresponding response with such expertise.

| Message # | Performative | Sender | Receiver | Content | Linked concepts |
|-----------|--------------|-------------|-------------|-------------------------|---------------------|
| 51 | Query | Operator | Operator(s) | Predicate: hasExpertise | NoteType |
| 52 | Inform ref | Operator(s) | Operator | Predicate: ownExpertise | Expertise, NoteType |

Protocol 6: querying about the recommended operation plan to the recommenders. This protocol is composed by 2 messages: querying an operator about its recommendation in the subtype of the desired asset, and the corresponding response with such recommendation.

| Message # | Performative | Sender | Receiver | Content | Linked concepts |
|-----------|--------------|-------------|-------------|------------------------|------------------------------|
| 61 | Query | Operator | Operator(s) | Predicate: recommends | NoteType |
| 62 | Inform ref | Operator(s) | Operator | Predicate: recommended | List of Operations, NoteType |

5. Beliefs, Plans and Goals of Node Agents

Agents representing assets are defined according to the next beliefs, goals and plans:

Beliefs:

- myOperator: the operator agent that performs maintenance operations (AgentIdentifier). It takes a value when the goal Achieve Assigned is fired.
- MyState: Time to the next required maintenance operation. This value is updated each time a maintenance operation is performed. This value is directly proportional to the quality of the maintenance operation plan quality and the expertise of the operator agent (both global and local). When such time is reached, the goal Achieve Updated would be fired.
- myType: description of the type and subtype of the asset represented by this agent. They are defined in the initial desire of the agent (Perform Setup) and they do not change along all the agent lifetime.
- Time: Current system time, used to compare its value with the one of myState. It is automatically updated each second.
- MySelf: identifier of the agent representing the asset (AgentIdentifier).

Goals:

- Achieve Assigned: This goal intends that the node agent would find out an operator agent. It is fired through an internal event launched when myOperator belief has no-value assigned. The plan SearchAssign would take place, instantiating the corresponding protocol 1..
- Maintain Updated: This goal intends agent node to be continually operative. It is fired through an internal event launched from the comparison between MyState and Time. The plan RequestUpdate would take place, instantiating the corresponding protocol 2.
- Perform Setup: This goal intends to assign the initial values of beliefs myType and myState.

Plans:

- Execute Plan: This plan is launched by an external event: the reception of a message of type request execution of operations belonging to protocol 3. This plan goes ahead with this protocol and computes the corresponding maintenance operation, modifying the values of myState and the expertise (global and local) of the operator agent.
- Request Assign Plan: This plan is launched by the goal Achieve Assigned. It executes protocol 1, giving a value to the belief myOperatori with the AgentIdentifier concluded after the execution of this protocol.
- Request Update Plan: This plan is launched by the goal Maintain Updated, when the value of the belief myState is less than Time, executing protocol 2, to request a new maintenance operation plan.
- Setup Plan: This plan is launched initially and it gives initial values to the beliefs myType and myState. e.

6. Beliefs, Plans and Goals of Operator Agents

Agents representing maintenance operators are defined according to the next beliefs, goals and plans:

Beliefs:

- **myNodes:** list of infrastructure elements that operates this agent. It is updated after each execution of protocol 1.
- **myType:** type (not subtype) of the asset in which is specialized this agent. It is decided initially in the desire Setup and it does not change along all agent lifetime.
- **myExpertise:** is a pair of global and local expertise. Local expertise shows how right the composition of the maintenance operation plan for the asset was. Global expertise represents how much general experience the operator agent has, it improves with practice. Both take initial values in the desire Setup, and they change after each execution of the protocol 3.
- **myOperations:** sequence of maintenance operations performed last time with each type of asset. It is a list of pairs (subtype, operation list). It is updated each time a new maintenance operation plan is composed combining the received recommendations (during the execution of protocol 3).
- **myRecommenders:** List of pairs (subtype, AgentIdentifier list) corresponding to the operator agents who to ask for recommendation about the maintenance of a asset. This list is computed during the execution of protocolo 4.
- **myRecommendations:** List of pairs (subtype, (recommender AgentIdentifier, recommender expertise, recommended operation list)). This list is computed during the execution of protocols 5 and 6.
- **MySelf:** AgentIdentifier corresponding to this operator agent.
- **MyGUI:** The graphical user interface assigned to this operator agent, where it is shown the plans that the agent is adopting, jointly with the received and sent messages.

Goals:

- **Achieve Register df:** This goal intends to register in the Directory Facilitator of JADEX at the begining of the agent execution, pointing out its specialized type of asset as service type.
- **Achieve Deregister df:** This goal intends to deregister in the Directory Facilitator at the end of the agent execution.
- **Query Recommenders:** This goal intends to know which other operator agents are specialized in the desired subtype of the asset. This goal is launched by Updating plan after the execution of protocol 2. This goal is linked with the QueryRecommender plan that executes protocol 4 with all the potential operator agents that were registered with the desired type of asset. The results were saved in the belief MyRecommenders.
- **Query Expertise:** This goal intends to know which (global and local) expertise have the operator agents that are specialized in the desired subtype of asset. This goal is launched by the Updating plan when QueryRecommenders has been satisfied. This goal executes as many instances of protocol 5 as recommenders were included in the belief MyRecommenders. The results were saved in the belief MyRecommendations.
- **Query Recommendations:** This goal intends to collect all the recommendations that the desired operator agents produce. This goal is launched by the Updating plan when the expertise of operator agents belonging to the belief MyRecommendations were collected. This goal produces as many instances of protocol 6 as recommenders were included in the belief MyRecommenders. The results were saved in the belief MyRecommendations.
- **Perform Operate:** This goal intends the execution of the maintenance operation plan. It is launched by the Updating plan when all the recommendations from other operator agents were collected. This goal produces the execution of an instance of protocol 3. The results were saved in the belief myExpertise and myOperations.

Plans:

- Request Operation: This plan is launched by the goal Perform Operate, that executes protocol 3. This plan implements the algorithm for combining plans from received recommendations, and updates the value of the belief myOperations, modifying also the beliefs myExpertise.
- Query Recommendation: This plan is launched by the goal Query Recommendations that executes protocol 6, and modifies the operation list belonging to the belief myRecommendations corresponding to the subtype of the asset.
- Recommending: This plan is launched by an external event, the reception of the message of querying recommendations that belongs to protocol 6. This plan goes ahead with such protocol sharing with the other operator agent the maintenance operations plan corresponding to the subtype of the desired asset saved in the belief myOperations.
- GivingRecommender: This plan is launched by an external event, the reception of the message of querying recommenders that belongs to protocol 4. This plan goes ahead with such protocol, sharing with the other operator agent the list of operator AgentIdentifiers that was saved in the belief myRecommenders for the corresponding subtype of the desired asset.
- GivingExpertise: This plan is launched by an external event, the reception of the message of querying expertise that belongs to protocol 5. This plan goes ahead with such protocol sharing with the other operator agent the own global and local expertise for the corresponding subtype of the desired asset that was saved in the belief myExpertise.
- QueryRecommender: This plan is launched by the goal Query Recommenders that executes protocol 4 and carries out the updating of the belief myRecommenders corresponding to the subtype of the desired asset, and the updating of the belief myRecommenders with the collected list of operator AgentIdentifiers.
- QueryExpertise: This plan is launched by the goal Query Expertise that executes protocol 5 and it performs the updating of the belief myRecommendations corresponding to the subtype of the desired asset.
- Updating: this plan is launched by an external event, the reception of the message of requesting updating that belongs to protocol 2. This plan goes ahead with such protocol and produces the sequential adoption of goals Query Recommenders, Query Expertise and Query Recommendations. After that, this plan adopts the goal Perform Operate.
- Assigning: This plan is launched by an external event, the reception of the message of requesting assignation of operator belonging to protocol 1. This plan goes ahead with such protocol and it carries out the modification of the belief myNodes with the AgentIdentifier corresponding to the asset assigned to this operator agent.

7. Execution of the JADEX agent system recommending plans

We have implemented and executed all these plans, beliefs, goals and protocols in order to have both operator and node agent executing. We will show in a sequence of screen captures the execution of both agents.

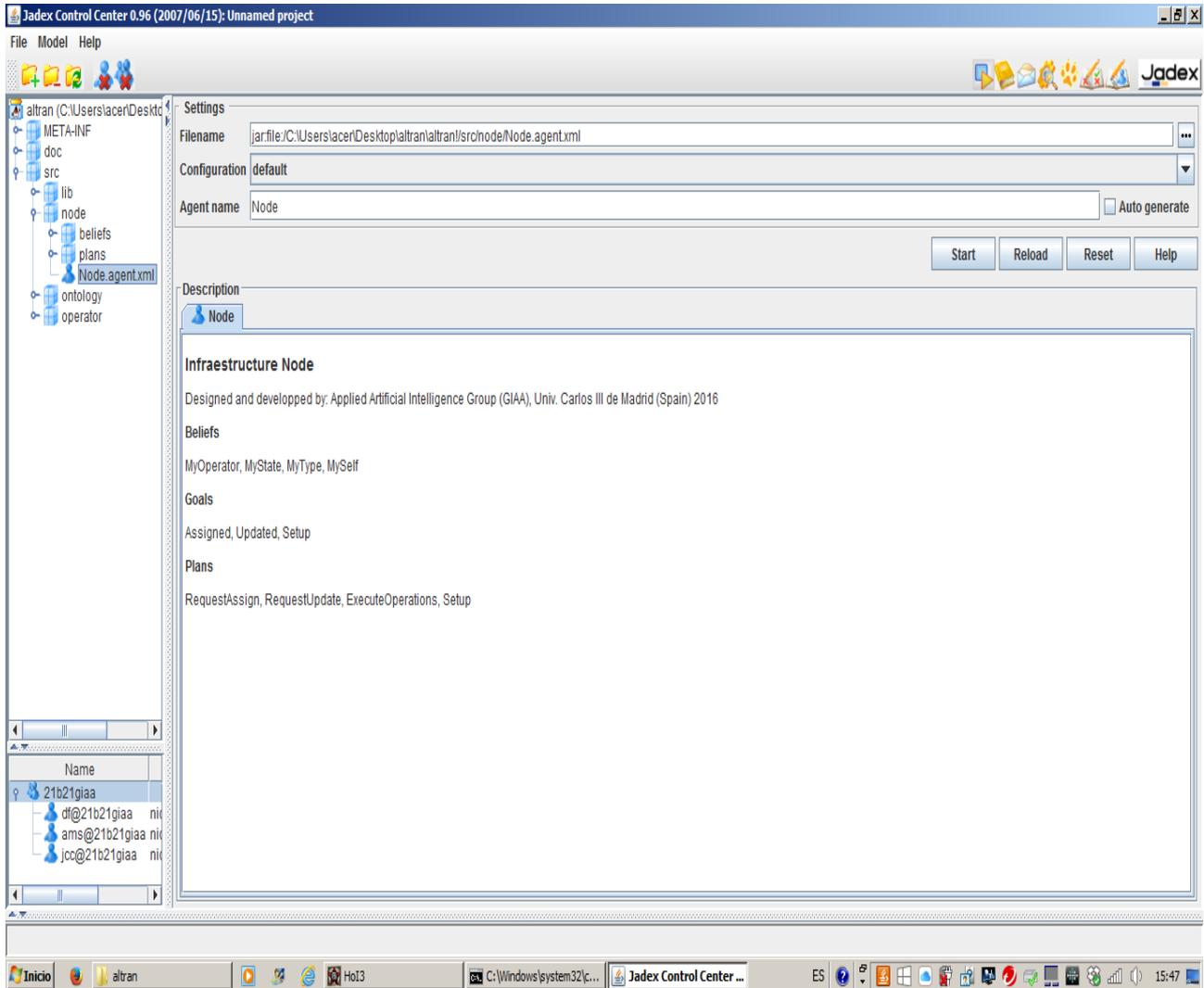


Figure 2. Node Agent loaded in JADEX platform

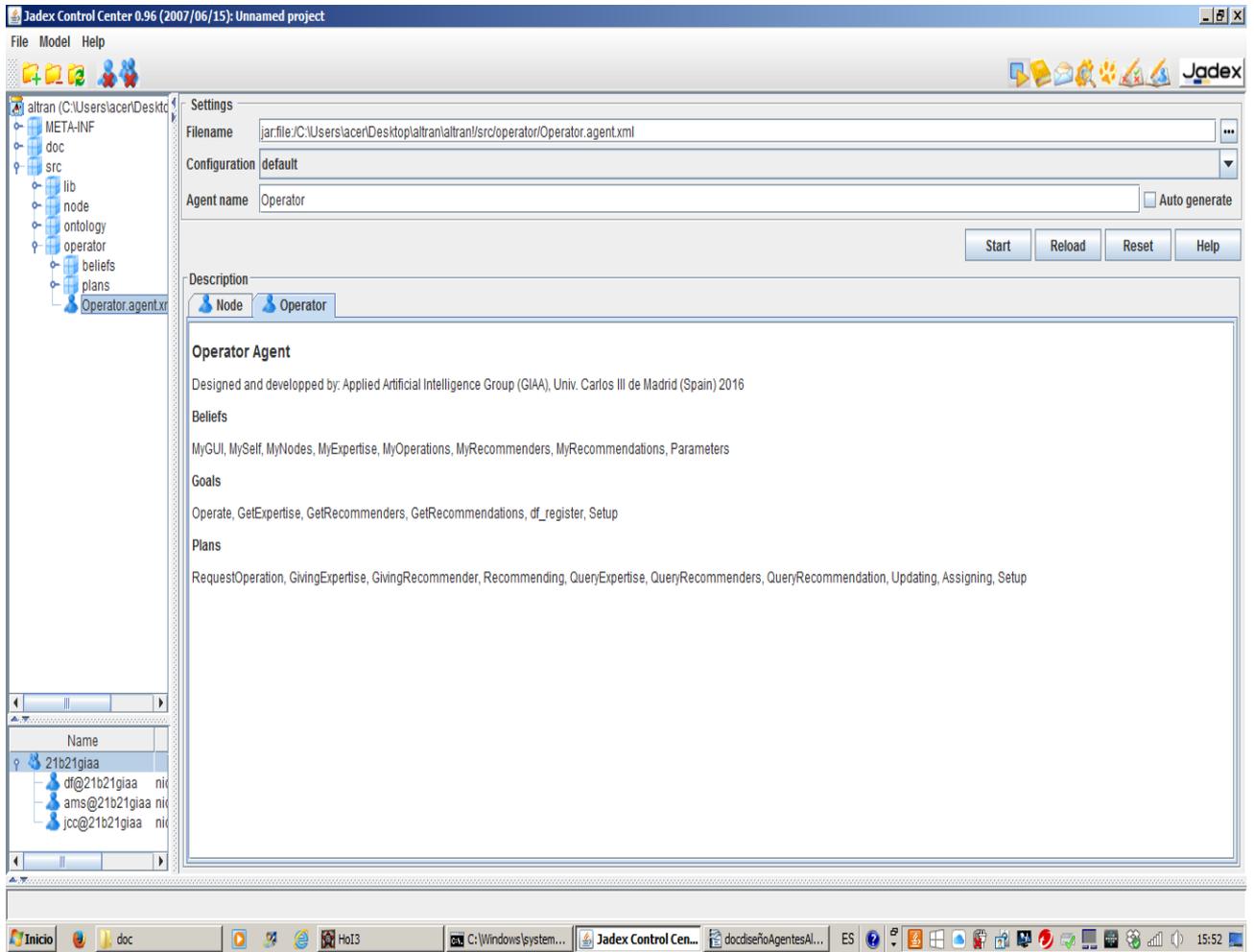


Figure 3. Operator agent loaded in JADEX platform

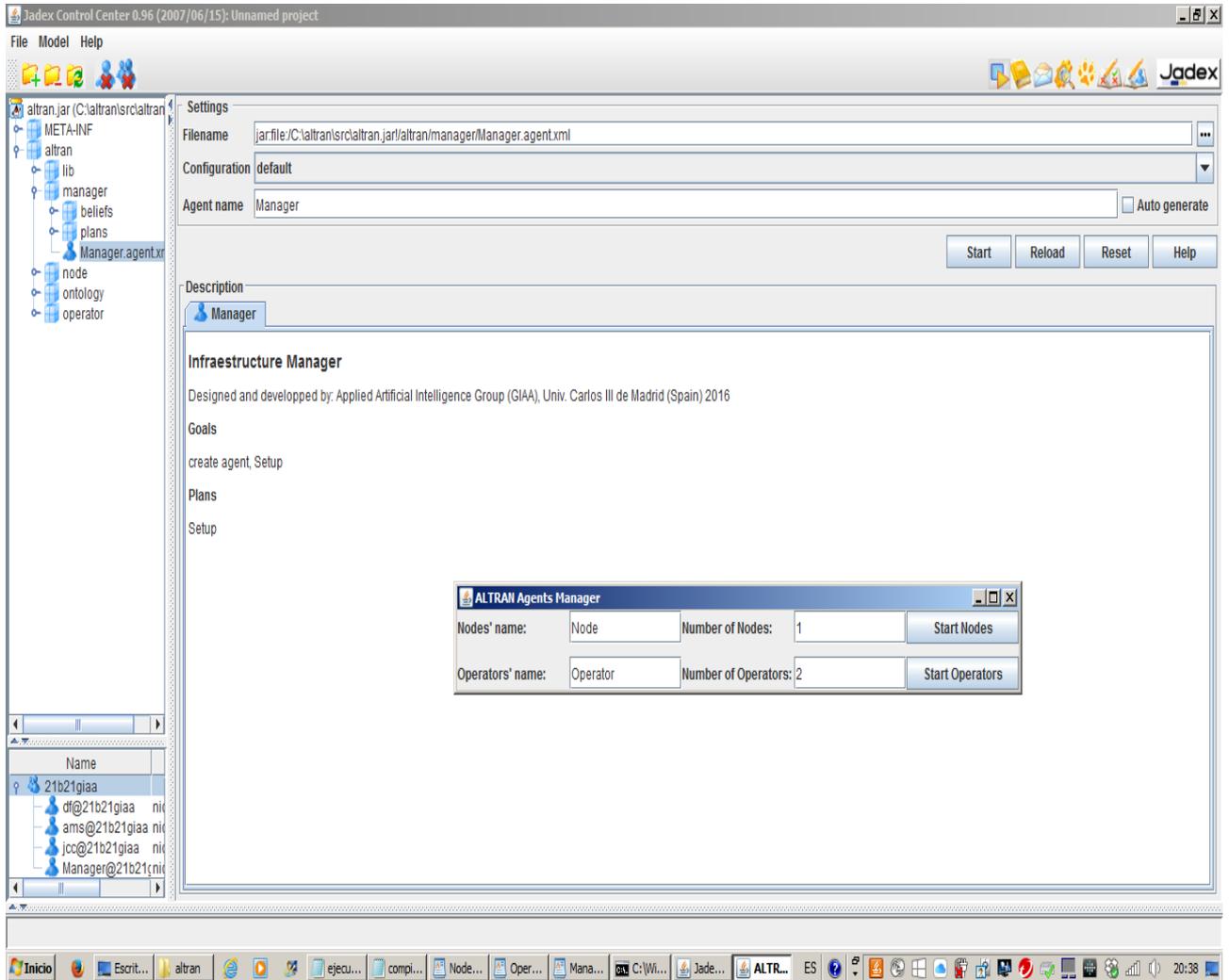


Figure 4. Selecting how many operator and node agents we want to launch

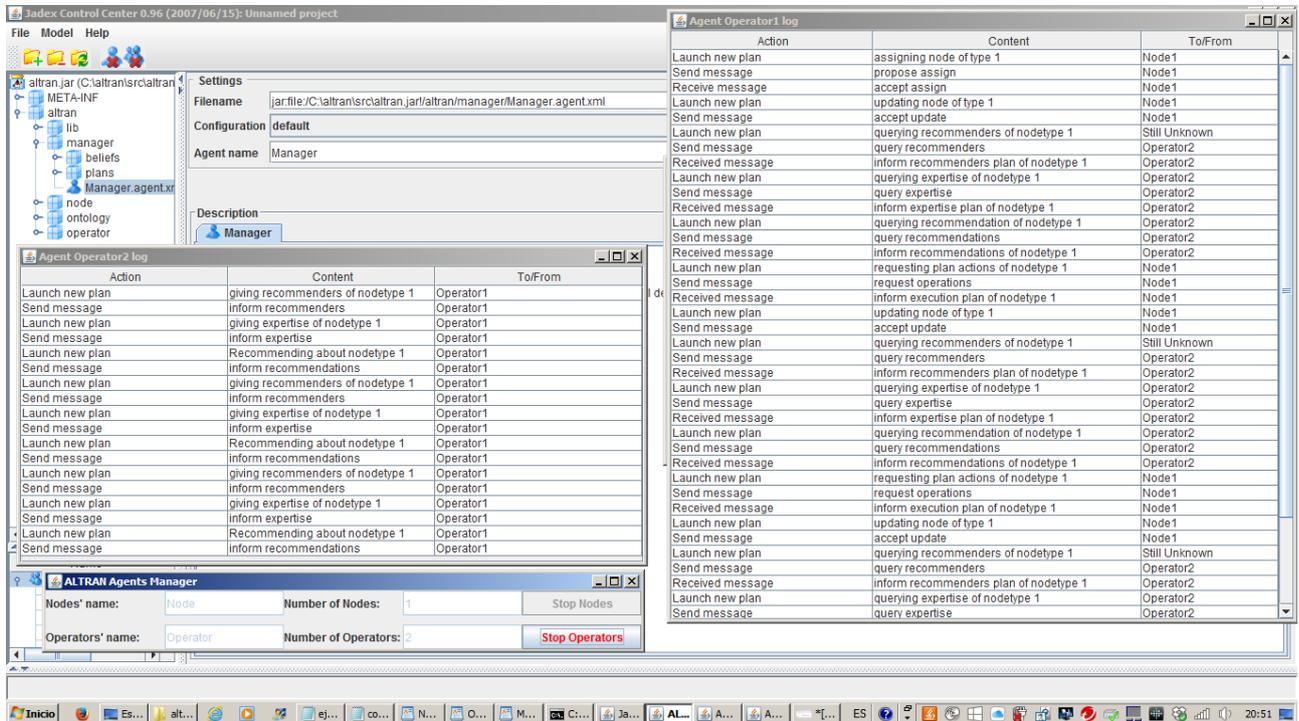


Figure 5 Operator and Node Agents running, adopting plans, sending and receiving the corresponding messages

8. Conclusions and future work

In this paper we have designed a new algorithm based on agent system theory, which is called "Cooperation on intelligent agent systems based on experience". The authors have developed new method to manage and capture the experience of more experienced staff in order to support other users with less experience in the asset management system. The new system has been developed using JADEX platform in order to facilitate the maintenance of infrastructures of an institution with several types of assets, and a high number of maintenance operators.

The right operation sequence that form a maintenance plan is unknown, as the possible existence of interrelationships between the operations. Such situation justifies the use of a recommendation system that would provide an useful plan to be applied over an asset generated from previous maintenance operations hold over other assets. The agent system we designed may act as a recommendation system providing a combination of already used maintenance operation plans. This combination takes into account the similarity of assets, the previous level of success of such plan (measured in how much time the asset required an additional maintenance operation plan), the general ability of the operator, and the expertise he has in specifically similar assets. Here we provide the full design of such agent system, including the ontology, beliefs, plans, protocols and goals of the two types of agents involved in the recommendation system. We also have shown how they effectively execute exchanging messages and adopting the corresponding plans. In future works, we intend to test different combination algorithms in order to show the relative pros and cons of using them.

In addition, Altran has deployed this algorithm in GAIA as a new innovative feature that will ensure the appropriated management of the assets' maintenance plans and, consequently, leverage this design to improve the efficiency of the global maintenance operations.

9. Reference

- Bellifemine FL, Caire G, Greenwood D (2007) *Developing Multi-Agent Systems with JADE* (Wiley Series in Agent Technology). Wiley
- Braubach L., Pokahr A., Lamersdorf W. (2004) *Jadex: A short overview*. In Main Conference Net.ObjectDays 2004
- Dennett, D. (1987), *The Intentional Stance*, MIT Press, Cambridge, Mass. .
- Ephrati, E. & Rosenschein, J. S. (1993), *Multi-Agent Planning as a Dynamic Search for Social Consensus.*, in Ruzena Bajcsy, ed., 'IJCAI', Morgan Kaufmann, , pp. 423-431 .
- Georgeff, M. P. (1983), *Communication and Interaction in Multi-Agent Planning.*, in Michael R. Genesereth, ed., 'AAAI', AAAI Press, , pp. 125-129
- Josang A, Ismail R, Boy C (2006) *A survey of trust and reputation systems for online service provision*. *Decision Support Systems* pp 618– 644
- Nwana H. S. (1996), *Software agents: An overview*. *Knowledge Engineering Review* 11, pp 205-244.
- Poslad S. & Charlton P. (2001), *Standardizing agent interoperability: the FIPA approach*. *Lecture Notes in Computer Science* 2086 Springer
- Rao, A. S. & Georgeff, M. (1995), *BDI Agents: from theory to practice*, in 'First International Conference on Multi-Agent Systems (ICMAS-95)', pp. 312--319 .
- Schneider, J., Gaul, A. J., Neumann, C., Hogräfer, J., Wellßow, W., Schwan, M., & Schnettler, A. (2006). *Asset management techniques*. *International Journal of Electrical Power & Energy Systems*, 28(9), 643-654.
- Sabater-Mir, J. & Sierra C. (2005), *Review on Computational Trust and Reputation Models*. *Artificial Intelligence Review* 24, pp 33-60
- Zhang, X., Arayici, Y., Wu, S., Abbott, C., & Aouad, G. (2009). *Integrating BIM and GIS for large-scale facilities asset management: a critical review*.