

# INTEGRATING A COMPETENCE MODEL AND THE MULTIAGENT SIMULATION OF VALUE ADDING NETWORKS

Min-Jung Yoo, Jean-Marie Fürbringer, Souleiman Naciri, Rémy Glardon

STI-IPR-LGPP

Ecole Polytechnique Fédérale de Lausanne

Station 9

CH-1015 Lausanne, Switzerland

{min-jung.yoo, jean-marie.furbringer, souleiman.naciri, remy.glardon}@epfl.ch

**ABSTRACT:** *Competence modelling and its simulation with enterprise Value Adding Networks (VAN) constitute an important issue that has not been widely studied so far. As a step towards the development of a generic interactive simulation platform for enterprise process optimisation and supply chain management, a competence modelling approach is studied and integrated simulation is performed on a multi-agent based simulation platform. This paper: (i) describes how to construct an enterprise competence model using a Semantic Web language and a rule-based system; (ii) shows how to integrate the competence conceptual model into the supply chain executable simulation model; (iii) discusses how such an approach can help appropriately allocating competences among different activities within a VAN and use efficiently the knowledge resources to increase the overall VAN performance and sustainability.*

**KEYWORDS:** *competence modelling, multiagent, simulation, value adding network (VAN), knowledge management, OWL*

## 1 INTRODUCTION

Today's enterprises are facing a continuous trend towards global operation and agile manufacturing. In order to remain competitive in an ever more demanding environment, they are concentrating on their main value adding activities. Thereby, they are increasingly integrated in complex Value Adding Networks (VAN) that are competing against each other. The design and optimization of these VANs constitute therefore a key issue in maintaining competitiveness. Due to the complexity of these systems, decision support tools based on discreet event simulation are often used for this purpose (Frayret and Glardon, 2007).

Competence management is known to be a key to success (Crozier, 1994). An adequate competence management leads to productivity improvements thanks to better motivation and stronger long-term work experience of collaborators whose tasks are clearly related to their competences (Csikszentmihaly, 1990, 2003). Within a VAN, competence management supports an appropriate allocation of resources and leads finally to a better cost versus performance balance.

The integration of competence management into the modelling and simulation of VANs constitutes therefore an important issue that has not been widely studied so far; few approaches have been proposed for competence management of enterprises in the context of global supply chain operation.

The difficulty in developing such an approach is due to the complexity of the modelled system: first of all, the methodology should support the conceptual modelling of enterprise competence which is in general very complex and large in scale; Secondly, the approach should be compatible with other enterprise applications and data bases; And thirdly, competences changing continuously with the collaborator's experience, the model state must be regularly updated. Additionally it must be said that the modelling of non-material resources such as knowledge, know-how and personal skills require sophisticated techniques that are very challenging for the research community but uneasy to implement in enterprises.

As a step towards the development of a generic visual interactive simulation platform for enterprise process optimisation and supply chain management, we have succeeded in developing a competence modelling approach using current open source APIs (Application Programming Interfaces) and integrating it into a multi-agent simulation platform.

The purpose of this paper is therefore to present this new approach, based on ontologies and multi-agent modelling, for the integration of competences into the simulation of Value Adding Networks.

More precisely, the paper:

- describes how to construct an enterprise competence model using a Semantic Web language and a rule-based system;

- shows how to integrate the competence *conceptual model* into the *executable simulation model* of supply chain;
- discusses how such an approach can help appropriately allocating competences among different activities within a VAN and use efficiently the knowledge resources to increase the overall VAN performance and sustainability.

From now on, the paper is organised as follows. In Section 2, a brief discussion of competence modelling and their importance is dealt with. Section 3 is devoted to the detailed concepts used in the competence system. After dealing with the integration issue of the competence model with the enterprise simulation model in Section 4, we present our prototype of the competence system and VAN simulation in Section 5. Finally in Section 6, the significance of this work and future directions are discussed.

## 2 COMPETENCE MODELING AND MANAGEMENT

This section presents the main competence modelling concepts that are implemented in the system. As these concepts have been published in previous articles, this presentation is restricted to a general description; the interested reader is referred to former publications (Pépiot et al. 2004, 2005, 2006 and 2007), (Fürbringer et al. 2005 and 2007).

### 2.1 The model

The competence model is imbedded in a model of the enterprise based on the activity (Figure 1). The *activities* are organized in *processes* and are associated to a *job*. An activity requires a given *capability* to be realized efficiently. The capability is an aggregate of competences and *resources*. A job is given to an *actor* who inherits a set of activities. The actor himself, being able to realize some activities, provides capabilities. A *gap* can then exist between the capabilities offered by the actor and the capabilities required by the set of activities attributed to a job. This gap depends as much on the individual competences as on the organization (meaning the distribution of the activities on the jobs).

*Unit competences* are characterized by *mastering levels* (Spencer and Spencer 1993). A mastering level specifies a degree of completion of an activity and of autonomy in a specific context. A mastering level requires specific resources that can be *cognitive* and/or *material*. The cognitive resources can be organized and/or evaluated in different ways. In several practical cases we made use of the *Bloom taxonomy* for this purpose. This presents the advantage of building a bridge with the training and education aspects of competence management (Anderson and Krathwohl, 2001, Kennedy, 2007). The material

resource refers to necessary (or available) documents such as norms, policies, certifications, copyright and even physical or virtual tools. Figure 2 shows an articulation of this model for a specific resource.

### 2.2 The uses of the model

Such a model can be used in different ways in a static human resource perspective, exploiting the coherence of the model and inferring information from a structured database:

- Who can realize a given activity?
- Who can occupy a given job?
- Who should detain a given capability, competence or resource?
- Which activities a given collaborator can realize?
- Which jobs a given collaborator can occupy?
- What is the gap between offered and required capabilities for a given staffing at the activity, process or enterprise level?
- What is the group of collaborators concerned by a given subject (resource)?
- What is the precise capability profile of a given job?
- What are the capabilities, competences and resources brought into play by a given collaborator in a given job?

Such uses are focused on the operational and tactical levels of competence management, supporting staffing, job definition, team management, recruiting and training.

On the strategic level, a competence model should support decision making related to competence distribution and development in order to assure robustness and efficiency. Using a criticality scheme combined with a SWOT analysis, an evaluation of the ability of a given competence structure to sustain threats and to benefit from opportunities can be performed (Fürbringer et al. 2007).

But the static perspective which is characterized by constant resources and constant demand is not able to take into account dynamic effects. Therefore, the integration of a dynamic perspective constitutes the main motivation of the work reported in this paper. By linking the competence model to an enterprise model we aim at developing a dynamic system able to answer questions such as:

- How robust is a given competence structure in a changing environment (resource and demand fluctuations, market evolution, technological innovation)?
- Which are the bottleneck capabilities, competences and resources (competence criticality)?

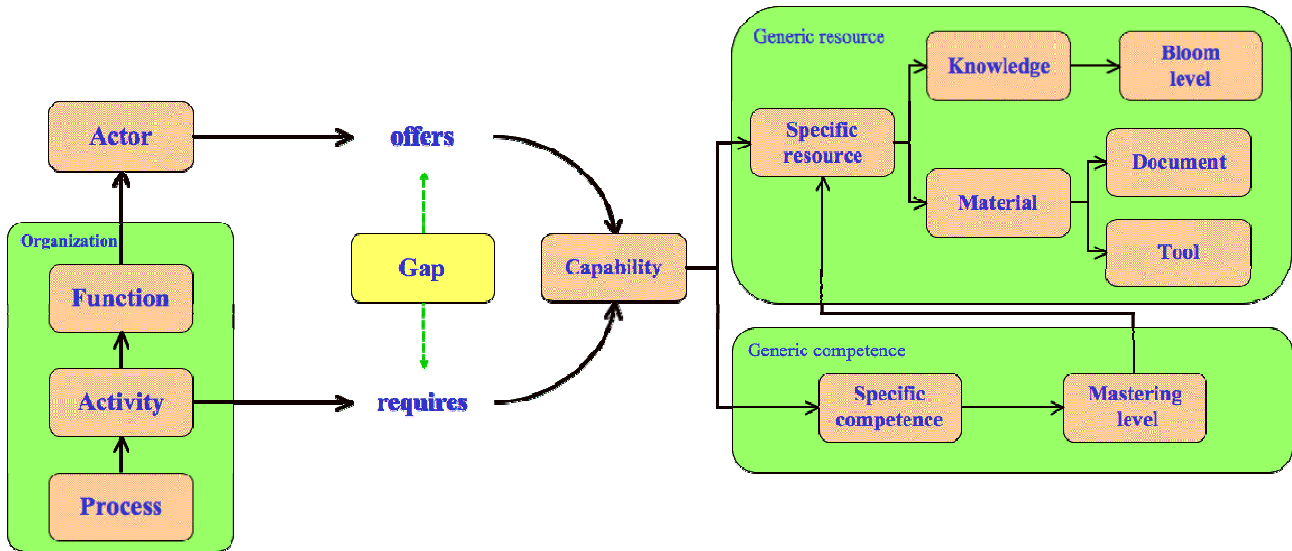


Figure 1. Competence and activity base enterprise model

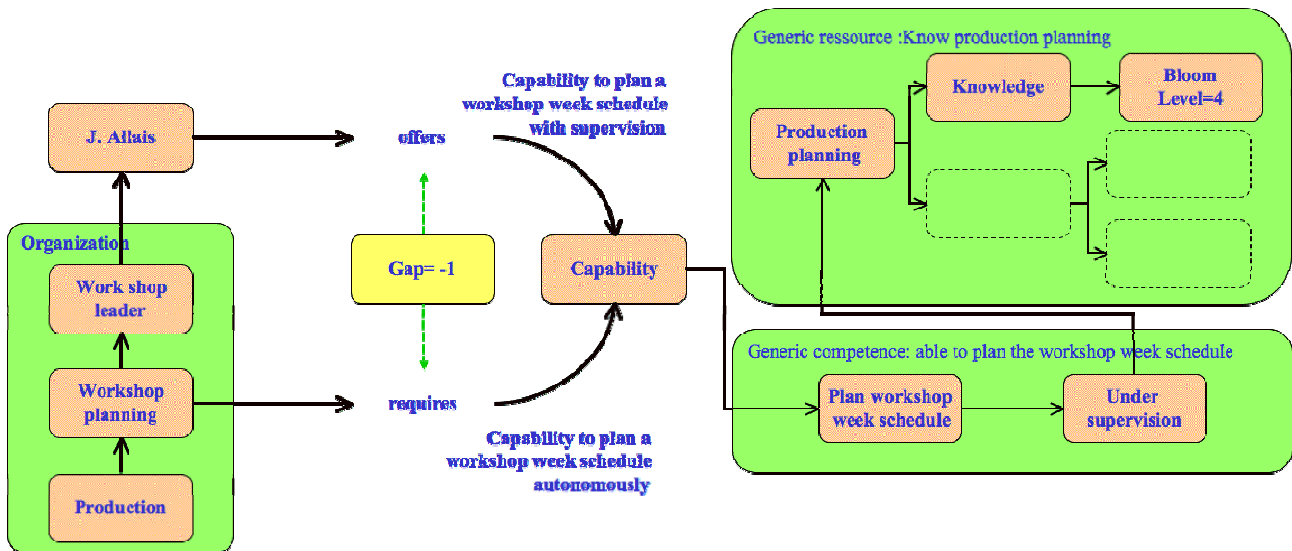


Figure 2. An instantiation of the competence and activity base enterprise model

### 3 SEMANTIC WEB-BASED MODELING AND RULE-BASED MATCHMAKING

In this section, we discuss the enabling technologies of the proposed competence modelling environment. The following subsections detail the chosen approach which is based on a rule based system and known Semantic Web languages.

#### 3.1 Semantic Web language for conceptual modelling of enterprise competence

Semantic Web languages such as RDF (Resource Description Framework, [www.w3c.org/rdf/](http://www.w3c.org/rdf/)), RDFS (RDF Schema, [www.w3c.org/rdf/](http://www.w3c.org/rdf/)) or OWL (Web Ontology Language, [www.w3c.org/owl/](http://www.w3c.org/owl/)) are largely used for various types of knowledge representation or ontology mod-

elling. According to (Daconta et al., 2003), the ontology level can be divided into several categories, considering the semantic richness, ranging from the taxonomy, to a thesaurus such as words and synonyms, a conceptual model such as RDFS, or to a logical theory - with very rich, complex, consistent, meaningful knowledge - such as OWL.

As seen in Section 2, competence modelling requires:

- the definition of words and synonyms, for instance “actor”, “function”, and “activity”;
- conceptual modelling, for example, “Paul” is an “actor”, “assembler\_1” is a “function”;
- logical rules such as “if an actor works for a function, it means that the actor must be in charge of an activity which is attributed to the function”.

Hence, a high level Semantic Web language is well suited for modelling enterprise competencies. Additionally, modelling enterprise competencies using semantic web languages brings the following benefits:

- Many open source tools exist for supporting knowledge resource design, such as graphic editors, model validation and inference tools.
- Semantic Web languages are open standards. Using such a standard makes it easier to share different knowledge bases among different systems and projects, either by importing other ontologies or by exporting one for other modelling purposes.
- The resulting model can be relatively easily integrated into other enterprise application contexts. Several vendors of enterprise solutions such as Oracle and SAP have already considered the use of RDF and RDFS languages within their new products.

Consequently, choosing a Semantic Web language for enterprise competence modelling is a promising approach in the context of competence management and VAN simulation.

Even though a model can be directly described in a language using a text editor, it might be better to use an editing tool providing an appropriate Graphic User Interface (GUI). The tool ‘Protégé’ was chosen for this purpose (Noy et al, 2001). Protégé is a free, open source ontology editor and knowledge-base framework. The advantage of today’s Protégé is the combined power of the plug-in components and the ability to tailor the environment by developing custom components (Henrik, 2003).

Moreover, using Protégé as a tool for modelling enterprise competence allows benefiting from the following plug-in tools:

- Protégé-Owl editors for the conceptual modelling and OWL description generating function;
- Generation of Java class definitions from the conceptual model: this function is useful for the integration with an executable simulation (see Section 5.3);
- Model validation functions (using ‘Owl model checking’ option);
- Class hierarchy visualization using OwlViz;
- JessTab (see Section 3.2 for more details) for integrating the Protégé-Owl model with an inference engine.

### 3.2 Jess and JessTab for dealing with the competence model

Jess is a rule engine (a software which applies rules to data) for the Java platform which is a reimplementa-tion of a subset of CLIPS (<http://www.ghg.net/clips/WhatIsCLIPS.html>). CLIPS (C Language Integrated Production System) is a forward chaining rule

language based on the Rete algorithm (Forgy 1982). The Jess rule language allows users to specify the rules to be modelled in a system. The Jess working memory contains a list of known *facts* and a set of *rules* upon which the Jess reasoner works to match them. While Jess can run as a standalone program without associating other Java applications, a more common use is to embed the Jess Java library into the Java source code of a system to be developed, and then manipulate the Jess library functions (provided with the ‘javax.rules’ package) from the system.

JessTab (link (a) on Figure 3) is a bridge between Protégé and Jess (Henrik, 2003). Basically, the Protégé-Jess integration allows developing graphically knowledge bases in Protégé and running problem solvers in Jess that uses Protégé knowledge bases to perform their task. By this way, Jess acts as the performance counterpart for Protégé while Protégé can act as the graphical knowledge-base counterpart for Jess in the context of conceptual modelling and its manipulation. If the Protégé model is embedded within the Jess environment, all instances in Protégé become ‘*facts*’ in Jess.

Some terminologies are now defined:

- We call ‘Competence Model’ the ontology, or the conceptual part of the competence representation.
- We use the term ‘Competence System’ for indicating the competence model and its rule engine counterpart, Jess (see Figure 3).

The discussion in this section concerns the competence model and its handling. The following section describes the integrate of such a competence system into a VAN modelling and simulation system for the purpose of analyzing to what extent the knowledge resources are allocated to and used for different activities.

## 4 INTEGRATING COMPETENCE SYSTEM WITH AN AGENT-BASED VALUE ADDING NETWORK SIMULATION PLATFORM

This section presents the basic idea of integrating two different model semantics: conceptual (competence model) and executable (enterprise modelling and simulation). The following sub-section handles in advance the notion of integration.

### 4.1 Taxonomy of competence integration

From our point of view, the integration of competence system with enterprise simulation can take place at three integration levels:

- Weak integration;
- Strong integration;
- Meta-level, or activity semantic, integration.

Weak integration of Competence System with enterprise simulation model means that, through the activity execution during the simulation, the system can calculate the statistics on using competences and knowledge resources. By observing the result, enterprise practitioners can analyze the appropriateness of current resource allocation.

Strong integration of Competence System means that the simulation takes the competence model into consideration as pre-conditions of activity execution. Therefore, before executing activities, the simulation system checks whether the competence and knowledge resources are satisfied in order to start a certain activity. If these conditions are satisfied, the simulation system executes the corresponding activities. If not, the simulation process might be blocked. By observing that all simulation processes are successfully executed, enterprise practitioners

can assure the appropriateness of current allocation of competences and knowledge resources.

Meta-level integration means that not only it is possible to check knowledge resources before activity execution, but also the competence system can be used as a means to finding pertinent activities according to a given product to be produced or a service to be offered. In that case, the activity element in the Competence Model should be rich enough to give full information on 'activity - product' relationship.

Our first prototype competence system is integrated into a multiagent platform for the simulation of a manufacturing enterprise's assembly activities considering the weak-level integration. In the following subsection, we give a brief description of the enterprise modelling and supply chain simulation.

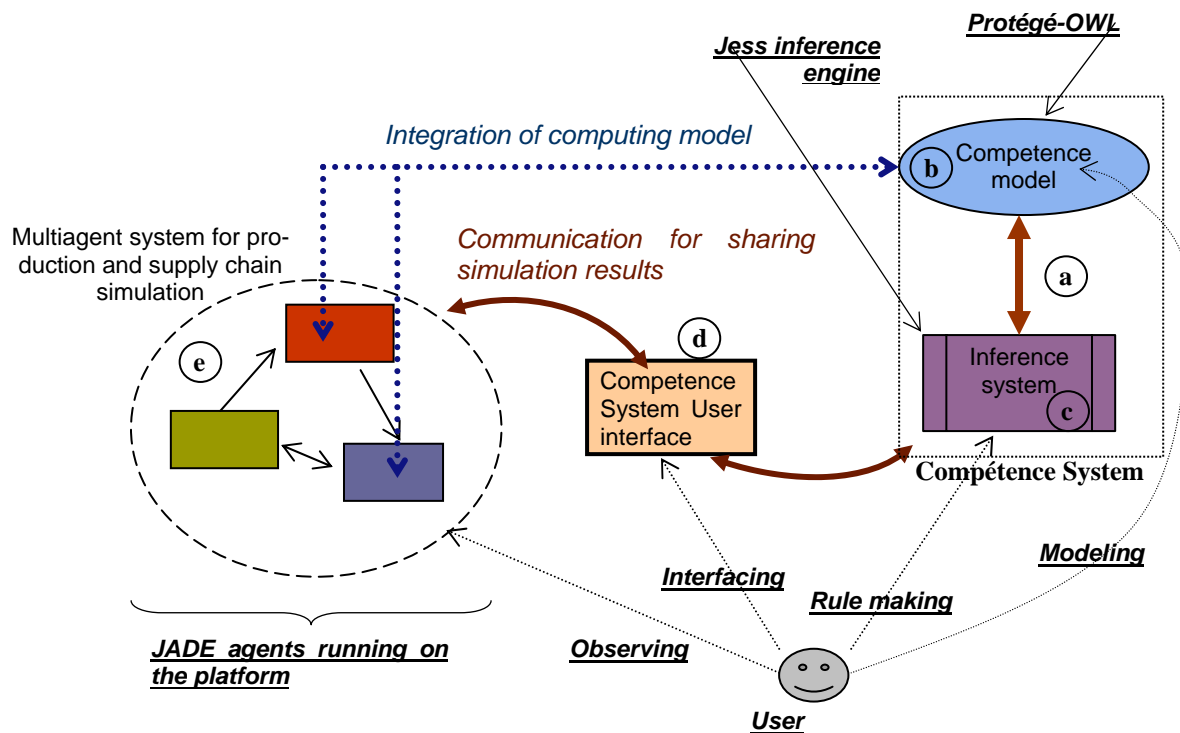


Figure 3 : Overview of the integration of the competence system into the VAN simulation platform

#### 4.2 Multiagent platform for enterprise modelling and supply chain simulation

As a full description of the agent-based VAN simulation platform is not the main focus of this paper, the following presentation is restricted to a brief discussion of the agent platform and of the currently available test version of an enterprise model.

The VAN simulation platform is based on publicly available open source agent libraries: Jade and Repast. Jade (<http://jade.tilab.com/>, "Java Agent Development

Framework") is a software framework implemented in JAVA to facilitate the implementation of distributed multi-agent systems. Jade meets a large part of the requirements thanks to the following characteristics:

- Jade agents are autonomous as they possess an object called "behavior" that guides their actions and reactions according to specific environment changes. This allows modeling supply chain activities and actors by gathering information, decisions and actions inside these "behaviors".
- Jade agents communicate using FIPA ACL semantics. This language pattern which uses performatives

such as “Request”; “Inform” and “Subscribe”, is helpful for structuring communication between agents and representing human communication in a realistic way.

- Jade is well-adapted for designing and running distributed applications. This means that agents can communicate in spite of their instantiation on different machines, running under different operating systems.

On the other hand, Repast (<http://repast.sourceforge.net/>, Recursive Porous Agent Simulation Toolkit) is an open-source software framework for simulation programming. The platform offers many interesting functions for supporting general-purpose simulation:

- Repast possesses an “Engine module” that contains, among several objects, an Engine Controller and a Scheduler (North and al., 2006). Controllers are used to start, pause, step the simulation, while the Schedulers control the flow of time. This last point is important for running the simulation in a realistic way.
- Repast proposes several tools for visualizing the evolution of the environment, of the agents and of some tailor-made performance criteria.

As a summary, Jade and Repast are complementary platforms as each one excels where the other one is weak. On the one hand, Jade offers a real agent structure relying upon a standardized communication while Repast does not give any framework for insuring communication consistency. On the other hand, Repast offers tools for discrete event simulation, batching modes and visualization whereas Jade does not guarantee process management and does not offer any user-friendly interface for simulation output visualization.

### 4.3 What should be measured?

The objective of integrating the competence system with a value adding network model is to study the pertinence of knowledge resource allocation among different activities within an enterprise. In order to achieve the weak-level integration presented in Section 4.1, two approaches have been tested:

- Integrating Competence Model and simulation
- Integrating Rules and simulation

The first approach is based on the following procedure.

According to the Competence Model defined in Sections 2 and 3, the actor competence appropriateness can be measured by comparing « required knowledge resource in order to achieve a certain activity and offered competences by actors who are realizing the activities» (“Gap” on Figure 1 and 2). However, the activity model can be considered in connection with the dynamic computational model, i.e., the notion of “behaviour” of the

agents. In other words, an activity “assembly pieces a,b,c” in the competence model can have an executable counterpart in a program, for instance, “call assembly piece procedure”

Each time the executable counterpart procedures are called from the simulation processes, the conceptual model parts are referred to for the purpose of retrieving corresponding information on *specific competences*, *specific knowledge* and *resources* (see Section 2 for their definitions). Through comparison between required competencies and offered competencies, a ‘matched competency use value’ is increased.

As for the second approach, it is based on a rule which refers to the current status of the ‘competency use value’. If this value is below a predefined level, then the rule is fired to provide information about the current situation. It makes it possible to users to observe whether a given situation occurs during the simulation.

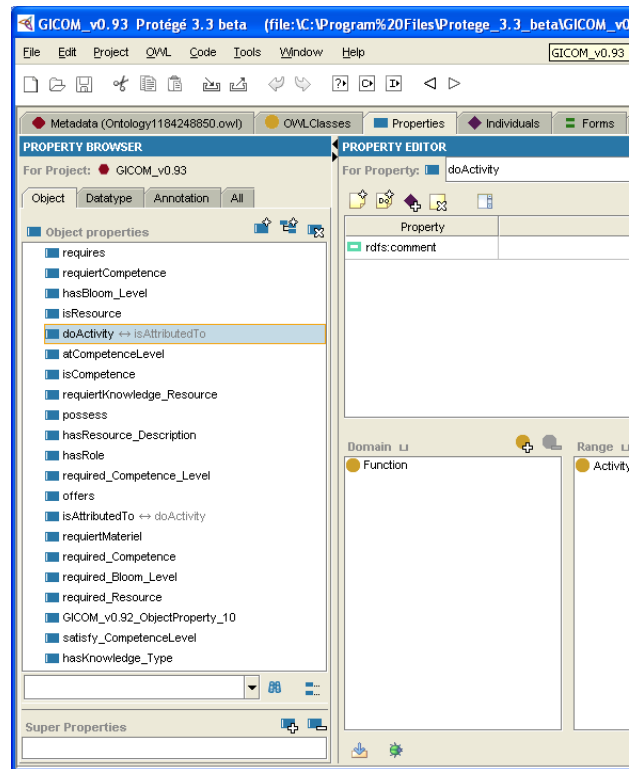


Figure 4 : Property list for the Competence Model

## 5 PRESENTATION OF THE PROTOTYPE

This section presents the prototype system in more details. Figure 3 shows the overall configuration of the competence system. The **competence model** ((b) on Figure 3) is designed using Protégé-Owl. The **inference system** (c) which contains facts and activation rules is developed using Jess.

The competence system user interface (d) is the entry point to the competence system, either for sending user

queries or for displaying the joint simulation results explained in Section 5.4.

Concerning the agent-based simulation part (e), each supply chain actor (e.g., branches, manufacturing enterprise) is modelled as “Jade Agent”. The VAN simulation platform can run either independently of or collaboratively with the competence system.

The following subsections present the design detail of each part.

### 5.1 The competence model: Classes, individuals and their relationships.

The current version of the generic competence model is composed of the following elements which are modelled as Protégé classes:

- Activity
- Actor
- Function
- Capability
- Generic Competence
- Competence Level
- Specific Competence
- Knowledge Type
- Generic Resource
- Bloom Level
- Specific Resource

In order to construct a specific competence model of a fictive manufacturing enterprise, other subclasses are added to the above-mentioned generic classes, specialising “Activity” and “Function” classes.

The “Activity” class has as subclasses “Assembly”, “Distribution”, “Fabrication”, “Management Planning”, “Sales”, and “Stock Purchase Management” which model different types of activities within a manufacturing enterprise.

As for the subclasses of “Function”, there are “Manager”, “Planning Employee”, “Production Employee”, “Purchaser”, and “Vendor” for modelling different functions within an enterprise.

The Protégé-OWL “*property*” element is used for explaining the relationships between individuals, such as “Activity A1 *is attributed to* Function Fc” or “Activity A1 *requires* a certain type of specific competence C2”. Figure 4 shows the list of properties defined for this purpose. The underlined expressions in Section 3.1 show some more examples of such relationships.

By creating instances of classes, for instance each collaborator, different functions and the relationships between ‘collaborator – function’, a complete knowledge base becomes available.

### 5.2 Inference System

The Inference System is composed of Jess functions and query definitions which can be used by a Java application to solve problems. The following description shows a part of the Jess program.

```
(mapclass Actor)
(mapclass Activity)
.....
(mapclass Function)
(mapclass Ressource_Specifique)
.....
(defquery search-person-function
  (declare (variables ?fname)
            ; fname is the center name (function) in string
            (object (is-a Collaborator) (:NAME ?name) (hasRole $? ?c ??))
            (object (OBJECT ?c) (:NAME ?fname))
  )
)
(deffunction get-person-function (?cname)
  (bind ?result (create$))
  (bind ?l (run-query* search-person-function ?cname))
  (while (?l next)
    (bind ?result (create$ ?result (?l getString name))))
  (return ?result)
)
.....
```

### 5.3 VAN and competence model integration

The enterprise and VAN models are composed of *basic elements* such as “Product”, “Material”, or *operational documents* as for example “Purchasing Order”, “Production Order”, “Assembly Order”. During simulation, different VAN actors communicate using ACLs according to the material and information flows.

The simulation model includes as well the following agents developed as “Jade agent structure” (Figure 5):

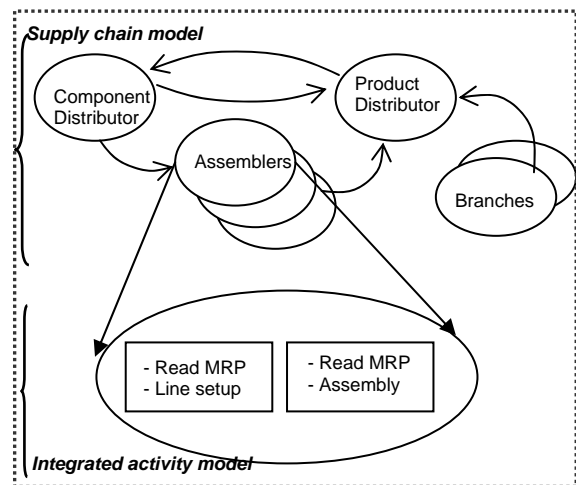


Figure 5. Supply chain model and activities in agents

- Agents representing enterprises, such as “Product Distributor”, “Component Distributor”, or “Assemblers”, in order to represent activities and business processes in detail.
- Other VAN actors: “Branch” agents for example.

These agents are monitored by “Jade RMA (Remote Monitoring Agent)” illustrated on Window (b) of Figure 6. The agent messages are traced with the help of “Jade Sniffer Agent” (Window (a) of Figure 6).

For the purpose of integrating an activity of the competence model into its counterpart in the executable VAN model, the following approach has been chosen:

- In modelling the activities within the “Assembler” agent, the notion of sub-functions (“assembly-line-1”, “line-septup-1”, “assembly-line-2”, “line-septup-2”, and so on) is implemented with their associated activities (“read mrp”, “line setup”, Figure 5).
- All these notions are defined as instances of “Functions” such as “PF-assembly” or “line-setup” which are subclasses of the “Assembly” class with corresponding competences and collaborators.

#### 5.4 Simulation with the competence system

The simulation starts with the start button that Repast provides ((e) on Figure 6). The Repast environment creates customer orders according to a given statistical model. The customer orders are then sent to the Product Distributor Agent and to the Component Distributor Agent in order to prepare the corresponding Assembly Orders. For the moment, the business processes within Product and Component distributors are simplified considering only the creation of *operational documents* and sending them to other neighbourhood agents using ACL messages. The inventory management schemes in each agent are relatively simple. The current status of Assembler is observable with the help of Assembly status windows ((d1) and (d2) on Figure 6).

Being provided with the competence system, users are capable of finding answers to questions listed in Section 2.2. The queries are sent from the competence system user interface ((e) on Figure 3), which then receives the query results.

During the simulation, while the Assembler agents are activated in order to execute their ‘behavior’, the above-mentioned activity methods are executed as agent’s internal actions. For example, if the system simulates the production of a final product during a certain period, methods corresponding to activities within the assemblers are executed.

The two windows (c1) and (c2) on Figure 6 are the competence system user interface which visualises the currently available actor resources at Assembler (see the “By Functions and Activities” tab on Figure 6). The offered resources by each actor can be shown using the tab “By Actors”. By observing the resulting statistics, the users can compare and analyse the following aspects:

- Whether all the resources offered by collaborators are efficiently used during the activity execution;

- Whether resources required by activities are offered by collaborators who are working for the function;
- If the required resources are not offered by any collaborators, find appropriate collaborators who might be able to offer the missing resources.

By reassigning actors to activities and re-executing the simulation, enterprise practitioners can increase the operational performances.

## 6 CONCLUSION AND FUTURE WORK

Competence management has become a key issue in assuring competitiveness on a strategic, tactical as well as operational level. Simulation is known to be a very powerful decision support tool, as it is able to reliably represent the dynamics of material and information flows. However, competence modelling is generally static and decoupled from any simulation model, thereby implicitly eliminating competence management from the simulation. The present work is an attempt to provide a solution to this missing link.

This paper presents an overall framework for the integration of competence modelling and enterprise simulation. The main following results are presented:

- Implementation of a generic competence model, using a Protégé-Owl ontology editor and its instantiation;
- Development of a Jess program which handles the above-mentioned competence model using its rule engine;
- Integration of the competence model into an already existing enterprise simulation platform

The feasibility of the proposed approach is shown with a prototype based on a fictive manufacturing company. It demonstrates the potential of integrating competence modelling with enterprise simulation. It shows that the following aspects need to be further developed:

- Improving the user interface by associating user-friendly GUI for creating competence elements and query handling;
- Achieving the other levels of competence integration: strong level and meta-level competence integration;
- Extending the competence model by associating the notion of “criticality” that will allow evaluating the robustness and sustainability of the enterprise with regards to possible external evolutions and internal resource changes.

The competence model described in this paper is related to a manufacturing environment. However, the authors are convinced that the proposed approaches are sufficiently general to be easily implemented in different con-

texts. In particular, successful applications of the competence model to service activities are an encouraging sign

for an extension to other domains such as health services, insurances or banking.

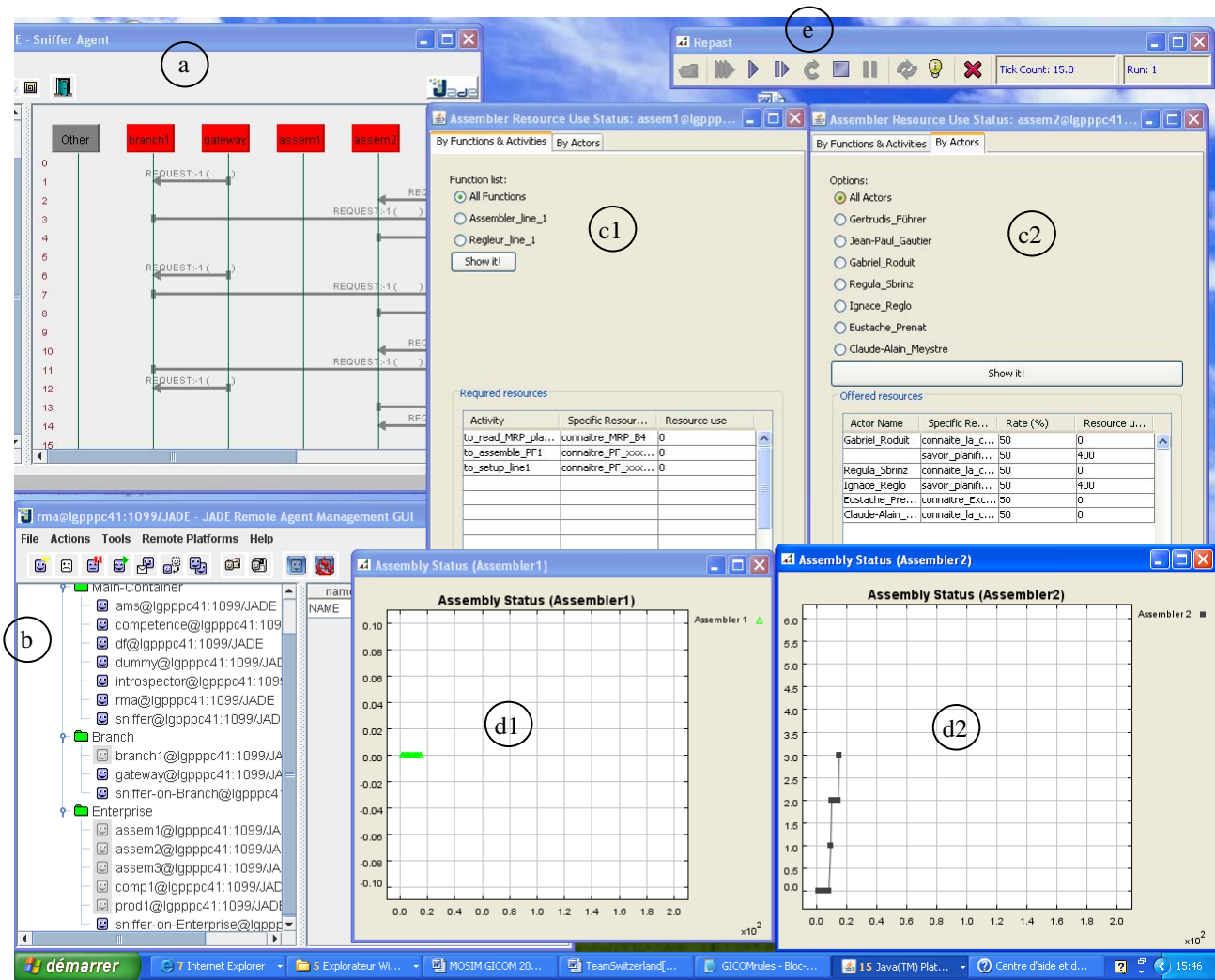


Figure 6. User interface of the simulation platform

**ACKNOWLEDGEMENT**

This work was supported by the Interreg Project GiCOM and partially supported by the IMS VIPNET project. Our thanks go to Professor C. Roche, Mrs. C. Million-Rousseau, and Dr. L. Damas from the University of Nancy (France), for their collaboration. The authors thank Dr. B. Fabio from the Telecom Italia for his helpful comments on Jade extension. Our thanks goes as well to Mr. M. Pouly, RCP of IMS VIPNET project, for his contribution to the project.

Jess and the Jess design are registered trademarks of Sandia National Laboratories. Jess was used in this work under an academic R&D license no 07-N06039.

**REFERENCES**

Anderson L. W., 2001. Krathwohl D. R. (ed.) et al, a taxonomy for learning, teaching and assessing. Longman.

Crozier Michel, 1994, L'entreprise à l'écoute:Apprendre le management post-industriel, point essais InterEdition, in fiche de lecture de E. Sohier, <http://www.cnam.fr/lipsor/dso/articles/fiche/crozier.html>, consulted on October 9th 2007

Csikszentmihalyi, Mihaly, 1990. *Flow: The Psychology of Optimal Experience*. New York: Harper and Row. ISBN 0-06-092043-2.

Csikszentmihalyi, Mihaly, 2003. *Good Business: Leadership, Flow, and the Making of Meaning*. New York: Penguin Books. ISBN 0-14-200409-X.

Daconta, M. C., L. J. Obrst, K. T. Smith, 2003. *The Semantic Web : A guide to the Future of XML, Web*

- Services, and Knowledge Management*, Wiley Publishing Inc.
- Henrik Eriksson, 2003. Using JessTab to integrate Protégé and Jess. *IEEE Intelligent Systems*, March/April.
- Frayret, J.-M., Glardon, 2007. Operations coordination in collaborative supply chains: state-of-the-art and research directions. *12th Annual International Conference on Industrial Engineering Theory, Applications & Practice*, November, 2007, Cancun, Mexico.
- Forgy, C. L., 1982. Rete : A Fast Algorithm for the Many Pattern/Many Object Pattern Match Problem. *Artificial Intelligence* 19, 17-37.
- Furbringer J.M., Glardon R., Glardon F., Pepiot G., 2005, Méthodologie COMET pour la gestion proactive des compétences. *Revue Française de Gestion Industrielle*, vol 24(2005), n°1, pp.117-127
- Fürbringer J.-M., Glardon R., Cheikhrouhou N., 2007, A methodology for managing competences within an organisation – A bottom-up approach, *19 ICPR Proceedings*, Valparaiso, Chili.
- Kennedy D., 2007. *Writing and using learning outcomes, a practical guide*. Quality promotion unit, University college Cork, Ireland.
- North, M.J., Collier N.T. and J.R. Vos, 2006. Experiences creating three implementations of the Repast agent modelling toolkit. *ACM Transactions on Modeling and Computer Simulation*, 16(1):1-25 Jan. 2006.
- Noy N. F, M. Sintek, S. Decker, M. Crubézy, R. W. Ferguson, M. A. Musen, 2001, Creating Semantic Web Contents with Protégé-2000. *IEEE Intelligent Systems*, March/April 2001.
- Pepiot G., N. Cherkrouhou, and R. Glardon, 2004, Modèle de Compétences: Vers un Formalisme, *Proceedings of the 5e Conférence Francophone de MOdélisation et SIMulation (MOSIM 04)*, Nantes, France.
- Pépiot G., 2005. *Modélisation des entreprises sur la base des compétences*. Doctoral thesis, EPFL, Switzerland.
- Pepiot G., N. Cheikhrouhou, J.-M. Fürbringer, R. Glardon, 2006. A fuzzy approach for the evaluation of human competences, *International Journal of Production Economics*. doi:10.1016/j.jipe.2006.08.025.
- Pepiot G., Cheikhrouhou N., J.-M. Fürbringer, R. Glardon, 2007. UECML: Unified Enterprise Competence Modelling Language, *Computers in Industry*, Vol 58, issue 2, pp. 130-142.
- Spencer S., Spencer M., 1993, *Competence at Work: Models for Superior Performance*. J. Wiley, ISBN: 978-0-471-54809-6.