Build Global System in Enterprise by Refactoring: Case Study

Nacira Ghoualmi-Zine, R.Boudour
Computer science Department, Badji Mokhtar University
Annaba Algeria
ghoualmi@yahoo.fr, R.boudour@yahoo.fr
Makhlouf Derdour,
C.U. Tébessa, AlgeriaComputer Science Department
m.derdour@yahoo.fr

***Abstract*-In this paper we present a study case concerning the integration in global system a set of information system in Algeria-Mittal steel enterprise. Study is based on three information systems: Sells management application, treasury application and turnover application. Communication system is a global system that integrates our chosen three information systems mentioned below. Communication follows rules of management imposed by the organism and it executes tasks defined by the managers. These applications are the most suitable example for us to implement our communication manager conceived in previous work [3][6]. We present methodology of refactoring to create conceptual network for each task. We present specific UML diagrams translated from generic UML diagrams developed in previous work. We present also the whole interfaces related to administrator and user. Refactoring in this case allows flexibility and adaptability of the enterprise and help it to modulate its fulfilling according to the constraints of the market: it integrate in such system new IS, suppress IS, create new task for user and build virtual enterprise and outsourcing.**

***Index Terms*- Global system, Integration, Interface, UML, Communicant entity, Conceptual/communicant network, Refactoring, Flexibility, Virtual enterprise.**

I. INTRODUCTION

Mittal Steel in Algeria is one of the biggest enterprises. Sites are dispersed and many applications are developed in several domains (Appliance, Commercial and Industrial Applications, Application Technology, Automotive Platform Managers, Design and Maintenance). The necessity of integrating the information systems/applications in a global information system is crucial. This must be coherent, reliable and follows the evolution of the organism. Yesterday, enterprises/organisms were efficient when they possess knowledge and produce it in masse. Today, they should offer communication tools and a fast access to information. The following notions appear zero delay, freshness of information, and distance works. Today all is shared, and virtual organism is in expansion. These decrease the cost of work and reduce execution time of the tasks. It also optimizes the useful of competence and resources (hardware and software)[1]. The conception of a communication system in an organism is very complex. It is involved in the majority of cases after the development of several IS applications. The communication system execute tasks that their execution implying a sub-set of IS/applications existing in enterprise. Communication depends on the type of task, the kind of the enterprise and others features like management rules. In previous work, we have developed a refactoring [2] [5] model based on the concepts of hyperisation. The result is a generic hyper system model allowing navigation among all components, which is the communication system. This paper is organized as**:** Section 1 presents Mittal Steel and the necessity to integrate applications in global system. Section 2 presents the set of information systems. Section 3 present the methodology applied to the set of information system and our selected two tasks. Section 4 present UML diagrams. Section 5 presents all interfaces of our prototype those for administrator and those for user. Finally section 6 is a conclusion.

II. INFORMATION SYSTEMS SET

In this section we describe the information systems set implemented in ALGERIA/Mittal steel. These information systems are the most suitable set for our study. There is a strong relationship between their functions, so we can generate conceptual network to execute distributed task. Each sub task is belonging to one information system. Thus we can obtain communication inter information systems.

A. Sells application

The goals of this application are to manage the sell of products. It is implemented in commercial office.

- The management of the commands (orders).
- The follow-up of stocks.
- The follow-up of expeditions.
- The establishment of order forms.
- The establishment of waybills.
- The establishment of invoices.
- The follow-up of the regulations (payments).
- The edition (publishing) of the various states.

B. Treasury application

The goals of this application are

- The treatment of invoices (supplier s)
- the treatment of transfer order
- the treatment of checks

C.Turnerover application

It I is a system developed to follow up the fiscal treatment of the studied enterprise

III. THE METHODOLOGY

It consists to apply the techniques hyperisation of a linear document. The result is a global system named hyper system allowing navigation among all components, which is the communication system**.** The level of granularity is [8] a module associated to an icon of the menu. With this model ISs/applications are decomposed into sofware entities or modules. They represent communicant entities. Thus, we propose refactoring of applications in term of modules (software entities) or communicant entities. The genearted entities are involved in the execution of tasks and constitute a network of cooperation(communication). Module can be involved in the execution of several tasks. Furthermore,the physical aspect of networks is negligeable in our investigation**.** Hyperisation gives us 28 entities (we can't present all entities but we present that are concerning task).

A. Selected communicant entities

- In first step we outline communicant entities

EC 1: Management of the sales of corresponding product (Bill/payment)
EC 2: follow up the turnover/ treatment/Follow-up turnover / treatment / Destocking of the product (Establishment of the list STK)
EC 3: Follow-up of the finance / treatment / invoice supplier
EC 4: Follow-up turnover / treatment / fiscal declaration)

- In second step we apply the predicate "execution task" to determine all communication networks [3]. In our study case we have two tasks. We present below interactions matrix for each one.

Task 1 (Destocking of the product (Establishment of the list STK)

	EC 1	EC 2
EC 1	1	1
EC 2	0	1

Fig.1. Interaction Matrix for task 1

So, communication network is:

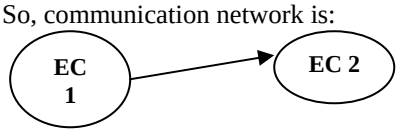


Fig. 2.Communication network for T 1

Circle represents entity and arrow represents direction of communication. Exchanged information could be messages or data, in our case it is data.

Task 2 (Fiscal declaration of turnover and edition)

	EC 1	EC 3	EC 4
EC 1	1	1	0
EC 3	0	1	1
EC 4	0	0	1

Fig. 3. Interaction Matrix of task 2

So, communication network is:

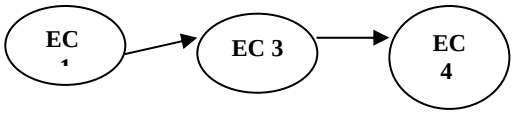


Fig. 4. Communication network of task 2

IV. UML MODEL

All UML diagrams [5] [6] are in French language because in Algeria it is the first language in economy.

A. Use cases

User:
Open session with: Identity control and Control Authentication, Execution task T1, Execution task T2, Close session

Administrator:

- Open session with: -Identity control Authentication Control
- Update Task base. Update communication network.

Update communicant entities base, Update user base and Close session

B. Class Diagram

Our system has 5 principal classes. Figure 6. below show relationship between classes and structure of each one.

C. Activity diagram

Execution task T1:

The activity of execution task T1 proceeds as follows:
1)The user asks for the execution of T1; 2)The system locates network RC1 of task T1; 3)The system invoke communicating entity EC1 (invoicing); 4)EC1 carries out the first sub-task (invoicing) which provides the result to the system, 5)the system memorizes the result;5)The system invoke communicating entity EC2 (Destocking the product (Establishment of form STK));
1)The system transmits the parameters translated to EC2; 2)EC2 carries out task T1 (Destocking the product and statement of stock STK) and provides

the result to the system 3)The system memorizes result; 4)The system transmits result to the user

Execution Task T2 :

The activity of execution task proceeds as follows: 1)The user asks for execution task T2; 2)The system locates network RC2 of the T2 task; 3)The system invokes communicating entity EC1 (invoicing); 4)EC1 carries out the first sub-task (invoicing) which provides the result to the system; 5)the system memorizes the result; 6)The system invokes communicating entity EC3 (treatment invoices supplier); 7)The system transmits the parameters translated to EC3; 8)EC3 carries out the second sub-task (payment invoices supplier) which provides the result to the system; 9)The system memorizes the result; 10)The system calls upon communicating entity EC4 (calculation of the taxable turnover and tax declaration); 11)The system transmits the parameters translated to EC4; 12)EC4 carries out the T2 task which provides the result to the system; 13)The system memorizes the result; 14)The system transmits the result to the user.

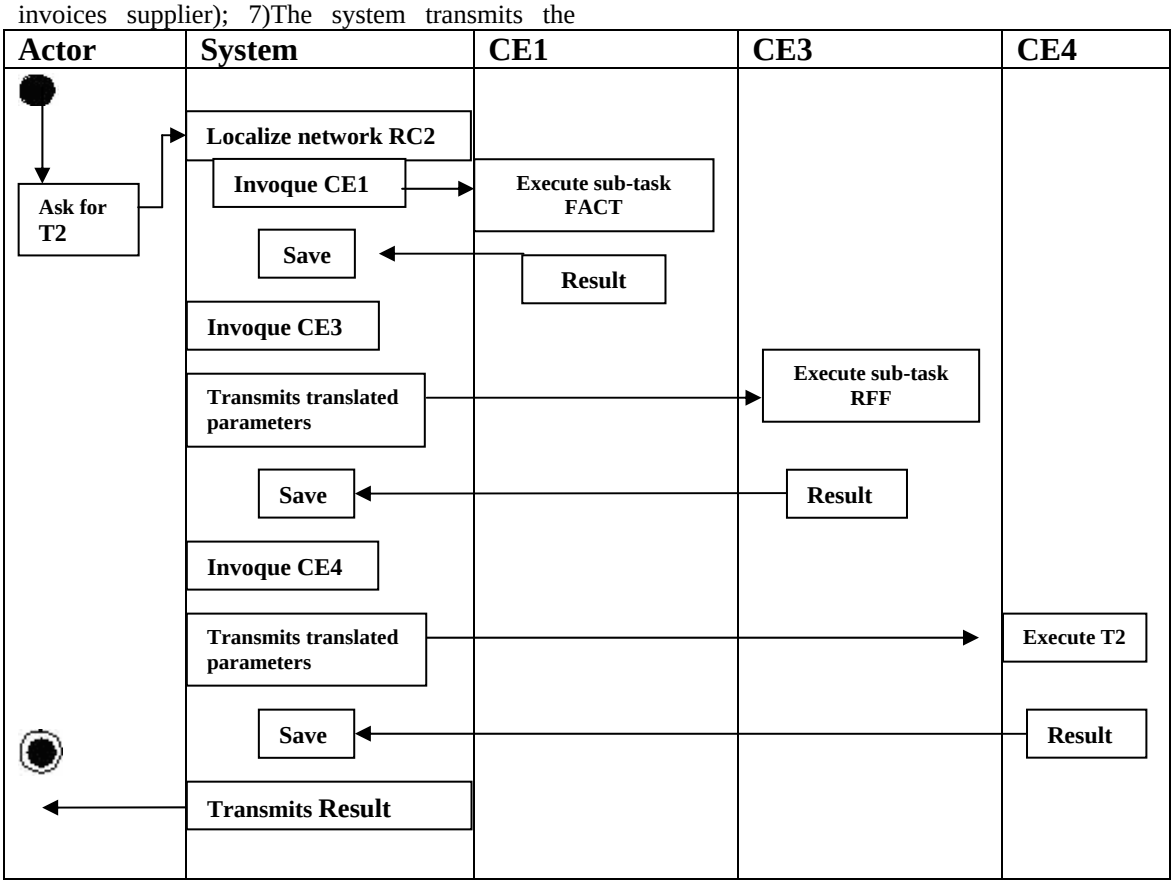
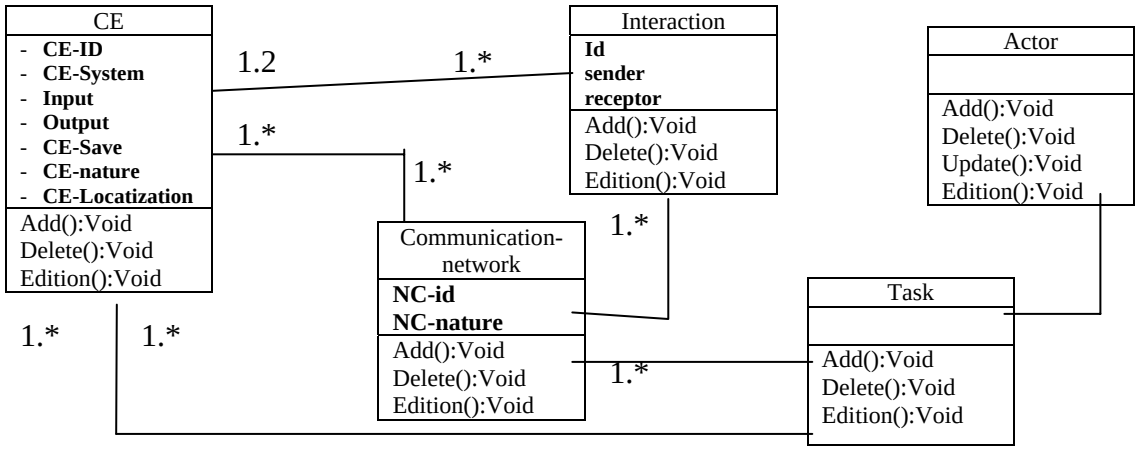


Fig. 6. Class Diagram(down)/ Fig. 7. Activity Diagram "Execution Task T2"(up)



V. GLOBAL SYSTEM'S INTERFACES

Administrator's interface is displayed including four button and two menus. To implement our system, we use JBUILDER.

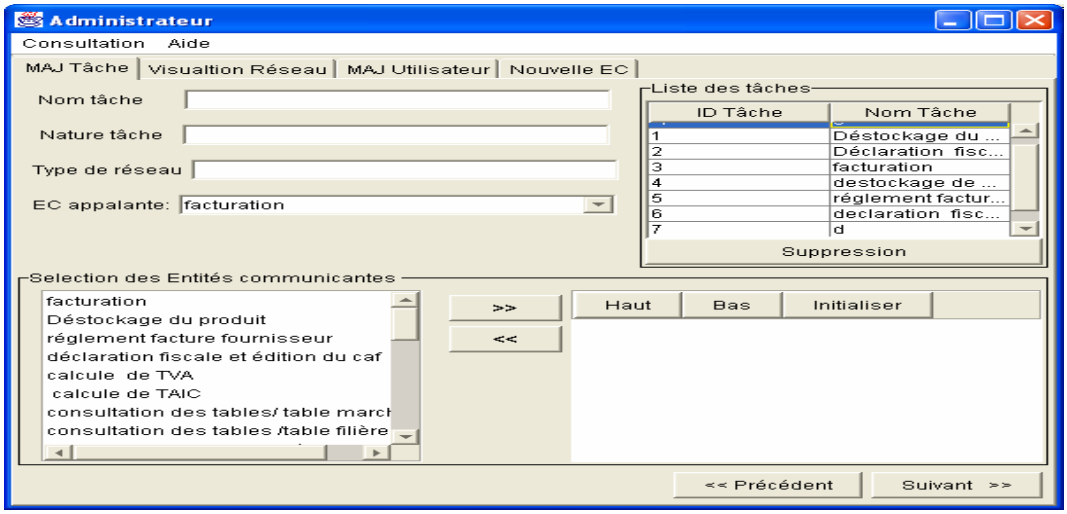


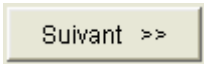
Fig.8. Administrator's menu's interface

A. Description of buttons

▪ **First button** "MAJ Tâche" gives the possibility to the administrator to add or to remove a task.
When the selected button is to add a new task, it proceeds as follows:
Input information of the task (Name, Nature, Type of network, caller EC)

- Select with 

communicant entities implied in this task

- Select with 

, Users allowed to execute this task**. Second button** "Network Visualization": allows the administrator to select a task and to visualize its interactions matrix.

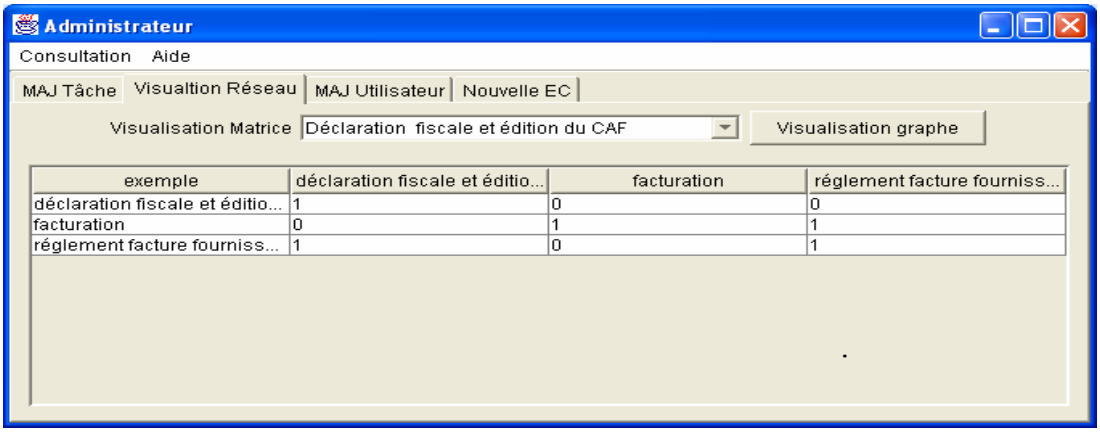


Fig.9. Interaction matrix associate to define one task

Third button "MAJ User", at this level, the administrator can:

- To add a user by filling the fields with information concerning the new user (Name, First name, Password, Function). Then, he selects with 

the set of task that this user is allowed to execute. At each task he validate with 

.

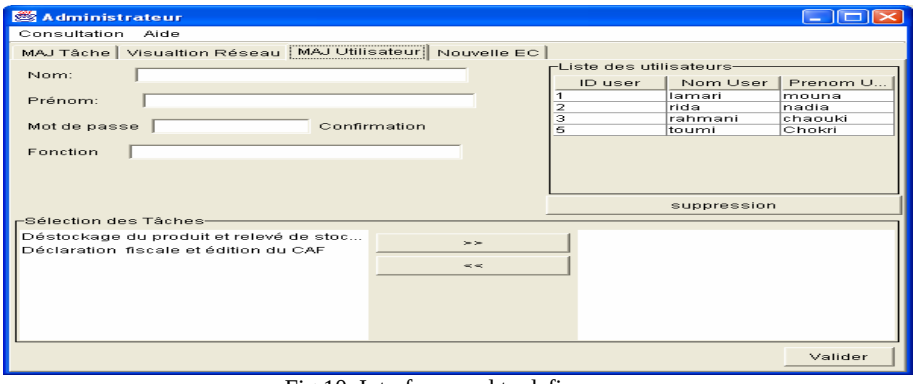


Fig.10. Interface used to define user

Fourth button "New EC.": permits to administrator to add a new communicant entity by introducing its information (Name entity, system entity, Nature entity, Localization, input data, output data)

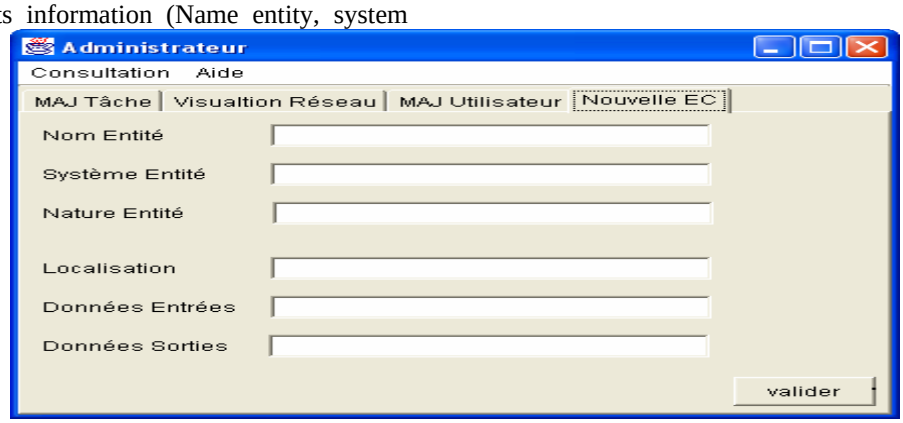


Fig.11. Interface used to define communicant entities

VI. Conclusion

In this paper we present a case study concerning integration of information systems in global system. We use refactoring to create communicant entities and network communication. At this step of our work, conceptual network are determined previously. So, communication manager select the network in network-base and execute the selected task. We use UML diagram to conceive our system and implement the communication manager with two types of interfaces: one for administrator and the second for user. Refactoring in this case allows flexibility and adaptability of the enterprise and helps it to modulate its fulfilling according to the constraints of the market: it integrate in such system new IS, suppress IS, create new task for user and build virtual enterprise and outsourcing. In future work we have to conceive no determined conceptual network. Thus the communication manager has to create dynamically the network at the moment that the user executes the task.

Acknowledgment: We have to thank Ms. Rida and Ms. Lamari for their effort and contribution to develop this system.

VII. References

[1] C. Cioffi, M. Panti, L. Penserini, L.Spalazzi, E. Tonucci, S. Valenti. An Agent-based Plateform for Federated Information Systems: Somme Design Issue. Proc. of the International Conference : on Software Engineering Applied to Networking and Parallel/Distributed Computing (SNPD'00), May 2000

[2] Opdyke WF. *Refactoring Object-Oriented Frameworks*. PhD dissertation, Department of Computer Science, University of Illinois at Urbana-Champaign, USA, 1992.

[3] N. Ghoualmi-Zine, A. Araar, 'Hyper system: A model for communication system', WSEAS TRANSACTION ON INFORMATION SCIENCE AND APPLICATIONS, Issue 1, VOL 1, July 2004, ISSN 1790-0832, pp. 399-404

[4] Conallen J.: Building Web Applications with UML. Addison-Wesley (1999).

[5] S. Demeyer, S. Ducasse, and O. Nierstrasz. Finding refactorings via change metrics. In Proceedings of OOPSLA '2000, ACM SIGPLAN Notices, pp. 166-178, 2000.

[6] N. Ghoualmi-Zine, A. Araar,'Towards a Communication System as HYPER-WARE',ANSINET/Information Technology Journal, volume 4, issue 4, 2005 ISSN 1812-5638.