Intelligent Agents to Human Resource Allocation in the Design of Chemical Processes

Beatriz López†, Arantza Aldea†, René Bañares-Alcántara‡
†Department of Computer Engineering and Mathematics
‡Department of Chemical Engineering
Universitat Rovira i Virgili
Ctra. Salou, s/n, 43006 Tarragona, Spain
{aaldea,blopez}@etse.urv.es, rbanares@etseq.urv.es

Abstract

The impact of information technologies, and particularly computer supported collaborative/cooperative work tools, has created great changes in large institutions that are now reshaping the way they manage and organise work in order to be more competitive. This paper presents the design of a multi-agent architecture that can help to organise a working team based on the characteristics of the type of problem to be solved and the features of all the possible people involved in solving this problem. These ideas will be applied to chemical process design.

Keywords: multi-agent systems, multi-agent team work, working team, chemical process design, people organisation, information systems, group behaviour learning.

1. Introduction

The impact of information technologies, and particularly computer supported collaborative/cooperative work tools, has created great changes in large institutions that are now reshaping the way that they manage and organise work in order to be more competitive. In the past, large firms, particularly process industries, used a hierarchical organisation approach in which a manager divided the workload among a set of workers who were told what to do (functional or departmental approach). Tasks were divided in such a way that each person was concentrated on the topic he was an expert on. That division caused the loss of the overall goal of the organisation activities; every department tried to do its work as much better as it can, without worrying on the correctness of the complete set of tasks [Alter99].

Nowadays, technology has evolved and provides professionals and engineers with more autonomy and high quality managers with more power. Consequently, companies are in the process of re-organising their workforce around a new process organisation paradigm to benefit from new technologies and to be more competitive [Daniels94, Hammer96, Alter99]. In this new organisation perspective, the selection of the working team to solve a given task, e.g. the design of a new chemical process, requires the analysis of the individual features of engineers, but also the consideration of other factors such as the ultimate goal, engineers communications, and technical issues. The new emerging technology can be used to study and analyse all these factors. Within the process engineering community there are several research groups working on process design. All of them are aware that the design evolves from a very abstract entity which encapsulate the objective and the constraints of the design towards more detailed designs of the different areas until the final artifact is created. All this process is performed by a team of engineers who work in different aspect of the design. The final artifact will be a composition of all this work. Due to the inherited distributed characteristic of all the processes, research has been carried out in helping the designers to represent all these process engineering models by using distributed architectures [Struthers98, Han95, Parunak98]. There is also research on how to keep all the information required to perform the design [Westerberg97] and how to maintain the deliberations and rational behind decisions [Bañares97].

Although researchers are aware that human experts perform the final design, little work has been done on how to manage the engineers that have to work together and cooperate and collaborate among them. In this research we propose a design of a system that helps the project manager to select the best working team available for the characteristic of the process to be designed.

The paper is organised as follows: next section is concerned with the problem of managing human resources in a design process where an overview of the process and the main criteria that can be utilised to decide the best working team are presented. Section 3 describes our proposed system which is based on a multi-agent architecture (MAS). Detailed aspects of every agents are described in section 4. Finally we conclude with a discussion and future work.

2. The decision making process: Managing human resources

The use of intelligent agents to help allocating task to people and form a working team is a difficult task. It is necessary not only to decide who is the best person to perform a task but also to foresee how all the team will work together and fulfil the final goal. Currently [Tambe97, Pynadath99] are developing a team-oriented programming toolkit that puts together diverse existing agents in a team to perform a task. However, the use of intelligent agents to model social and skill behaviours of human expert within a working team is quite a novel approach in which multi-agent theories must be interrelated to social science models of human behaviour.

In the particular case of chemical process design, a qualified manager is required to study the objective, goals and the tasks to be performed. This manager decides who are the best engineers to be recruited to perform such a task with the optimal outcome. To perform such management task the manager relies on the social, professional, and personal skills of every expert in the firm. Several factors are analysed in the decision making process. Some factors are engineer-specific (related to the expert that will perform the task) and other are domain-specific (constraint imposed by the own feature of the task to be performed). Finally, the experience of the manager solving previous similar design problems is essential to elaborate the adequate working teams.

Engineer specific factors that may be considered to model the skill of an expert in performing a specific task are:

- Capabilities. The type of task that the engineer is specialised on, what is his/her background.
- Experience. The experience is measured with a positive parameter in terms of the belief in performing a good job, but has a negative outcome as experience is expensive and to recruit such person will cost money. On the other end of the experience spectrum is an assistant (apprentice) that is cheap but also less reliable.
- Personality: social preferences, working hours, working methods, and others.
- Past experiences in chemical design, working teams, etc.

Domain-specific factors are those imposed to the chemical process that determine the importance and criticality of the whole process of a specific area or aspect of the design. Hence aspects such as: cost, safety, sub-processes involved should be considered. Global criteria may also be considered to form the working team.

Finally, in the decision making process, past experiences should also be analysed. If a team of engineers worked well in the past for a similar design there is no reason to believe that they will not work well again.

The selection of the team is then a difficult decision making process that involve many factors. Once a decision has been made the design process will start, and the project manager should monitor the performance of the engineers and be alerted in case that unexpected situations arise. If the team does not perform as expected, the manager should dynamically create a new working team taking into account the new situation. The manager deals with the whole design process and may dynamically adapt its working team plan due to unforeseen situations. In real life unexpected situations occur frequently, new constraints are imposed to the artifact, changes in the environmental laws, or even people cannot be available when we need them. Hence, the manager take control of the situation and explore new alternatives.

Note that at the current stage of our investigation we are working on the selection of the designers team and the simulation of how the team will interact among them; we are not actually concerned about doing the design of the chemical process.

3. Our approach

The benefits of the selection of an adequate working team has been exposed above. Several factors should be taken into account by the project manager to form a working team. To help a human resource manager in its decision-making we propose a multi-agent architecture where intelligent agents give advice about the best working team for a given design problem.

Multi-agent architectures allow the modelling of social behaviour of different human experts. Every agent corresponds to a human resource involved in an area or an aspect of the design process. We may have therefore three kind of agents:

- Engineer agents (EA) specialised in the design of a type of process (reaction, separation, heating...), environmental engineers, or process engineers. Note that there are several agents that potentially can perform the same task but they have different degrees of experiences and also different personalities.
- Technician agents (TA) who give support to the manager agent and the engineers agents. There is a directory of engineer and technician agents where the name, address and capabilities of each agent is registered.
- The manager agent (MA) whose main goal is to form clusters of agents (working team) in order to solve a given design problem.

Note that the multi-agent architecture serves as a platform to the whole design process. At the current stage of our research, we are dealing with the agents that allocate human resources. However, the same agents can be extended in order to provide a toolkit to help engineers along all the design process.

4. The Multi-Agent System

The multi-agent system receives a new design task (goals, constrains and objectives) and outputs a team of agents representing the human resources that can perform the task according to the task specification (see figure 1).

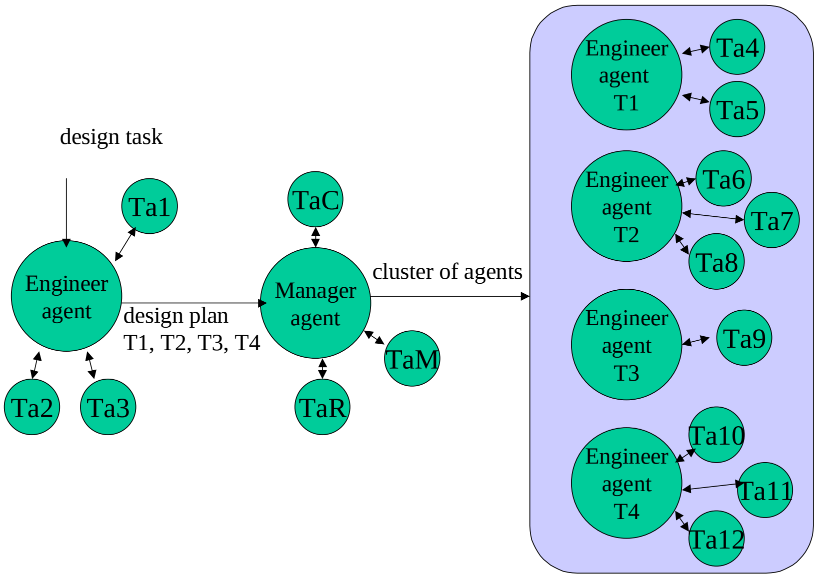

Fig 1. The engineer agent accepts a design task and produces a design plan. The manager receives the design plan and selects the set of agents that will execute the plan. Technicians agents (TA) help manager and engineer agents to perform their reasoning process.

We adopt the abstract refinement model for a design process proposed by [Han95], and the principles-based task decomposition as a systematic way of decomposing a design task into a set of subtasks. According to the principles-based task decomposition, tasks are defined at different level of abstraction. A task at level i can be decomposed into different subtasks at i+1 level. Then, an engineer agent that receives a task of level i may perform the decomposition of the task into its subtask by conforming a plan that is send to the task manager agent. At the higher level, a task is a simple design problem that can be accomplished by the knowledge of the agent (figure 2).

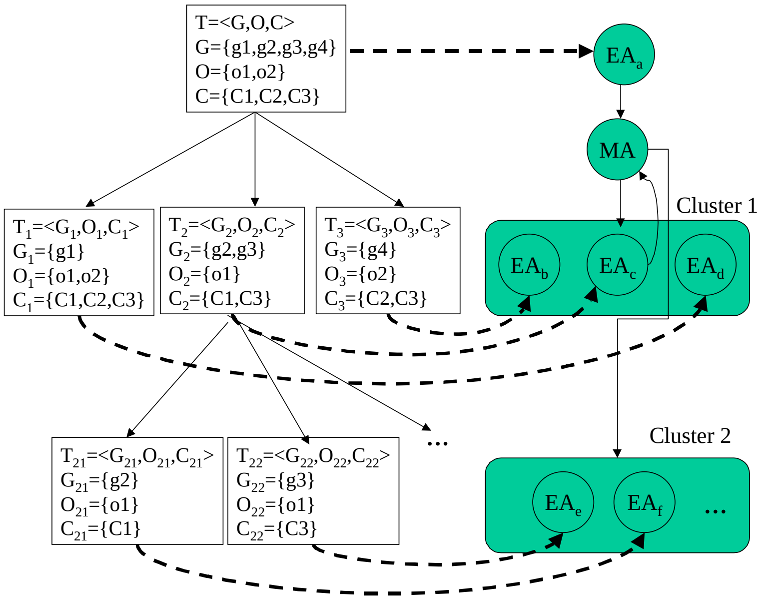

Fig. 2 Abstract refinement model and the corresponding working teams.

For example, let us suppose that we need to design a butene alkylation process. To do so, our system will receive the main goal, objective and constraint:

G: Design the butene alkylation process
O: To obtain the optimal design in terms of cost
C: Costs of the input streams and compositions
T: Design the butene alkylation process

At the first level, there is a single task "design the butene alkylation process" which is assigned to the engineer agent with project manager expertise (EA_a in figure 2). Such engineer builds a design plan of three tasks:

T_1: Initialise the project's specifications.
T_2: Synthesise the plant process flowsheet for the whole plant designs
T_3: Evaluate process flowsheet and create improved designs

Then, the Manager builds a cluster of agents (cluster 1) that will perform every task of the design plan.

Agent EA_c realises that task 2 is quite complex and it will need help for several other agents to perform that task. Hence EA_a decides to divide the task into subtasks (design plan), $\{T_{21}, T_{22}, T_{2n}, T_{2F}\}$, where:

T_{21}: Synthesise the plant complex structure and divide plant into subplants
$T_{22..N}$: Design every subplant (initialise, synthesise and evaluate). This subtasks will also be divided into subtasks if more than one EA is required.
T_{2F}: consolidate the flowsheet of all the subplants into one

And then, EA_c requests the Manager agent to recruit the agents that will perform each task of the design plan (cluster 2).

4.1 Engineer agent

An engineer agent is a model of a human expert, that is, a design engineer in our application. An engineer agent receives the request of a design task and can do one of the following activities: 1) propose the division of the design into several subtasks within a design plan[1]; and 2) study the effort of the design task when the task is not decomposable. In the former, we say that the engineer is acting as a group coordinator (see [Aldea00] for details). In the later, the engineer agent commits the task by offering the cost (feedback to the MA). The EA takes into account the fact that the cost of a re-engineering task will be less than the cost of a new, never seen before design. The memory of past design kept locally on the EA plays, then, an important role in the estimation of the cost.

4.2 Manager agent

Given the division of the design into several subtasks and knowledge about the design capabilities of the engineers, the manager agent decides to recruit some specific engineers agents (working team). Knowledge about the engineer agents came from three sources: rules, models and a cases. To reason about every kind of knowledge, the MA is aided by technician agents.

When the MA uses models and rules, it acts as a broker [Maturana99]. From the directory of agents, where every engineer agents has been subscribed by informing about its capabilities, the MA decides which agents recruit for a given task. The selection of agents is based on the contract net protocol. First, MA broadcast all the agents susceptible of performing a given task according to the directory. Engineer replies consist of the cost of performing the task. Among all the bids returned, the MA selects the one that fits better according to some predefined parameters of the task (e.g. cost, criticality, etc.).

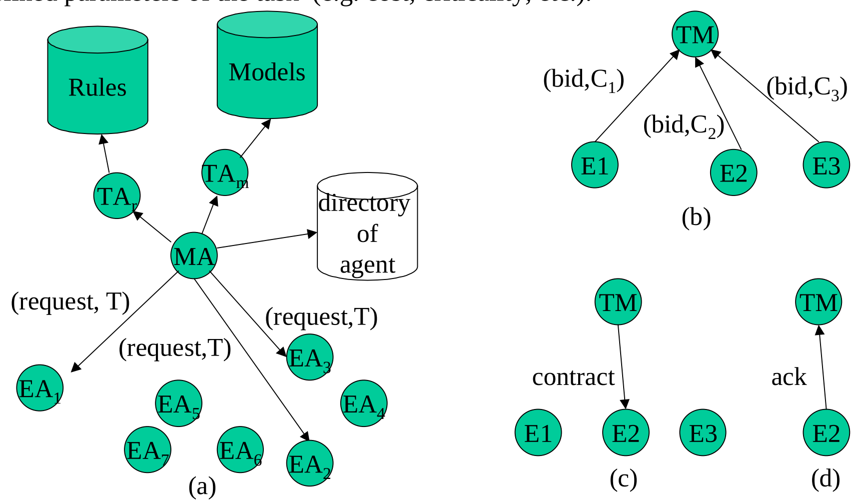

Fig. 3. (a) The manager agent acting as a broker: it sends a request to the agents that, according to the agent directory, can solve a given task.

For example, suppose that MA needs to assign a separation engineer to the "T_{221}: design the liquid separation subsystem" (figure 3). Fist MA looks up the directory of agents the ones that are separation engineers (EA_1, EA_2, EA_3). Then the MA requests to all of them for the cost of developing the task. Each of the separation engineers answers the request with the corresponding cost (C_1, C_2, C_3). Then, MA decides upon one of them by using heuristics (rules) and models. The EA_2 separation engineer is selected, and then contracted.

Costs are tuples that summarise processing time, cost of the action, confidence on the success of the action, and other useful factors of the requested task. Costs of the tasks can be simple or aggregated. The former is the case when an engineer agent is able to perform the design task without more decomposition (highest level of the abstract refinement model). The later is a consequence of the clusters of agents. If task T is decomposed into the subtasks $\{T_1 ... T_n\}$, then the resulting cost of task T is the aggregation of the individual tasks $F_{ag}(C_1, C_2, ... C_n)$ [Ribeiro96, Torra98]. Aggregate functions are not simple, since it depends on the parameters of cost: some can be added as for example money; others can be overlapped as for example time; and others should be combined according to some fuzzy relation, as for example quality. Costs of tasks are

[1] Note that the tasks of a design plan are not necessary sequential; they should be taken as a set.

propagated from the MA to the engineer agent that sends the design plan, and from the engineer agent to the MA that requests the task, until the first engineer agent receives the final cost.

When MA uses past experiences, that is cases, it falls back on the case-based reasoning technician agent in order to find the most similar situation in the past that resembles the current one. Then, MA requests the same set of agents.

Once a cluster is constituted for any of the methods, the MA keeps in its case memory, information about the design task (goals, constraints and objectives), the corresponding cluster and the final parameters achieved (costs, etc). Any event that arises during the design process is also registered in the case. For example, it can be the case that the clustered suggested by the multi-agent system, does not work as expected. Such information is a crucial feedback for the system improvement since it can learn from its failures. When such a situation is detected, the case is kept as a preventive situation: the multi-agent system will avoid suggesting the same cluster for the same design goal. Group behavioural leaning is then accomplished by both, learning from success and from failure.

When some unexpected situation arises, we can manually inform the system about the changes and the system will take control of the situation and explore new alternatives. For example, in case that a team mate fails, the system can ask to another engineer agent from the same cluster (working team) to assume its work. In case that the engineer does not accept, then the MA will look for other engineer available in the system.

4.3 Technicians

These are agents that will help the manager and the engineers with some particular aspects indirectly related to the design, likewise other technician staff aid human engineers during the design process. Therefore, they are specialist in a very specific aspect of the design that help the engineers to evaluate the cost of a solution, or may be agents that simulates a design alternative to verify that fulfil all the requirements. In our MAS, they are implemented as specific artificial intelligence tools as for example case-based reasoning to reuse previous designs, model-based reasoning to check the validity of a design, and others.

Note that a technician agent is attached to either the manager agent or a single engineer agent since they keep the specialist knowledge and reasoning of agents. Technician agents of a given agent, however, can share information (cases and models) can communicate each other in order to improve its behaviour (figure 4). For example, the model-based reasoning TA of the manager can help the case-based reasoning agent to test the validation of the current case. Particularly, we have introduced a knowledge discovery technician agent able to learn some relevant data for the case-based reasoner in order to index cases.

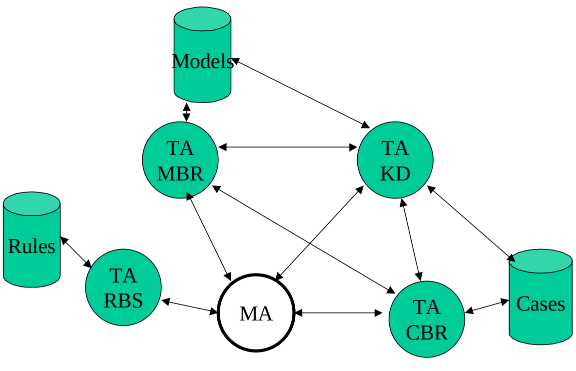

Fig. 4. Technician agents that helps the manager agent:
Model-based Reasoner (MBR), Knowledge Discoverer (KD),
Case-based Reasoner (CBR), and Rule-based Reasoner (RBS).

4.4 Agent Coordination

Coordination is based on the individual agent's agendas that locally keep information about the sequence of tasks to be performed by every agent. Each entry in the agenda represents a task, with a label, a description, the address of the agents which share the same design objective, and its preconditions (previous tasks that must be completed prior the starting of the task and its inputs). Coordination is the achieved by contracting, multi-agent planning and negotiating.

Agents are contracted to perform specific tasks. The manager agent is the responsible of contracting as explained in section 4.2, and the collaboration of the agents responds to the status of their agendas.

Multi-agent planning is performed in two ways: centralised and decentralised. When a task needs to be divided into subtasks, a centralised planning is performed: the MA contracts agents by assigning them individual tasks. Decentralised planning is performed locally among agents of a working team. Each agent has its own planning capabilities to organise its agenda. When no coordination activities are taking place, the agent continues working with the next task of the agenda. When starting a new task the agent checks if the preconditions of the task are satisfied. If they are not, the agent sends a message to every agent of the cluster informing about its unexpected situation. Then, the responsible agents inform about the their current status, the messages are sent to the agent that requests the information and the MA, so the later can decide what to do in case of a failure (see figure 5 for an example of the decentralised coordination planing which follows the extended UML methodology for agents [Odell00].) Alternatively, if the preconditions are satisfied, the agent will proceed with the task. Once the task is completed, the agent will inform the rest of the agents, so the other mates can start the tasks depending on this one. Agents, then, can optimise their activities by demoting tasks whose preconditions are not yet satisfied and promoting tasks ready for execution.

Finally, negotiation holds when conflicts arise due to unforeseen situations (changes in the design constraints, failure of a team mate, etc.). Conflict resolution between agents of the same cluster is solved via negotiation protocols directed by the MA (see, for example, the cfp of [FIPA97]).

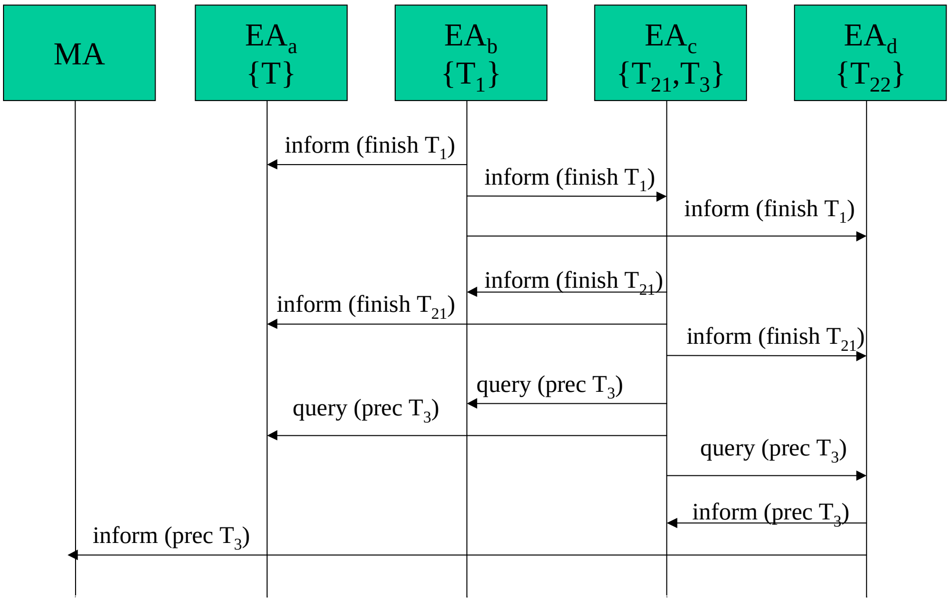

Fig. 5. Coordination among agents EA_a, EA_b, EA_c, EA_d, which perform task T. Note that task T is divided into T= $\{T_1,T_2,T_3\}$, and T_2 subdivided into $T_2 = \{T_{21}, T_{22}\}$, and T_2 must be completed before starting T_3.

4.5 Agent cloning

At any moment, a new engineer agent can be added to the system, as a new engineer can be incorporated in a firm. That can be the case when a new specialist is required, as for example, an environmental expert.

A new engineer agent is a clone of the previous ones: same structure, techniques, ontology, communication protocols, etc. The difference among agents raises in their specialist knowledge acquired from experience. So, at the beginning it is a novice, since it has no experience in the organisation. Technician agents that represent formal knowledge about design and compiled heuristics gives support to the new agent. Then, technician agents are also cloned when creating a new engineer agent [Struthers98].

Either engineer or technician agents, once created, remain in the system forever. The knowledge derived from the design process is kept locally in each agent, so it can be used for re-engineering.

The manager agent can also be cloned when the current MA is busy. That is, at any given level of the task refinement model, an engineer agent can request the MA to assign a cluster for the design plan produced. If the MA is busy, it can decide to clone itself. Note that the case-memory of the two MAs will be the same, as well as the rules and models. When a cloned MA finishes its goal, it is removed from the system. The case generated is kept on the case base and from now on it will be handled by the original MA. By cloning the MA we avoid bottleneck situations due to a centralisation system as the ones observed in [Pynadath99].

5. Discussion and Related Work

This paper presented the first outline of a multi-agent architecture that supports the management of human resources for the design of chemical processes. The system will be developed using team-oriented programming tools similar to [Tambe97] in which different heterogeneous agents are working together to fulfil a plan.

The multi-agent system is innovative in the sense that take advantage of the social activity of agents to learn models of group behaviour that are then used to give support to the human resource allocation. Group behaviour leaning is still a novel and challenge topic in the AI community. Robot soccer research has provided insights of how teams works that now are being extended to other disciplines [Tambe00]. Our approach follows the same ideas but it does not suppose a prior model of working teams; instead, our system will learn such group behaviour models. Models of behaviour are provided according to the current staff. In a dynamic company, where professionals come and go, the availability of a tool that adapts working groups to such changes is important.

One of the great advantages of multi-agent systems is that we can accelerate the group behaviour learning by simulating the agent society. This is one of our current research issues (see [Aldea00] and [Prietula98]). Another important feature in our system is the flexibility. That is, when building a working team the system does not search for engineers that exactly match a set of parameters [Pynadath99]; instead, the best team of engineers is selected according to a decision making process.

As a future research, one of the challenges is the fact that all the experts involved in the design of a process use different vocabulary, even language, since they can work in different places and in different countries. Some of the works regarding ontology's generation such as [Simeoff98] can be a good starting point.

Finally, note that the system proposed is the first step of the design process, where the working team is selected. The ultimate goal is to build a complete multi-agent system for the whole design process. In such a system, the monitoring of the working teams would be performed by the manager agent, and so alert and act on time when unexpected situations arise.

References.

[Aldea00] A. Aldea, B. López, R. Bañares-Alcántara. "The Use of Intelligent Agents to Manage Human Resources in the Design of Chemical Processes". Accepted to the Third Catalan Conference on Artificial Intelligence (CCIA2000), October 2000.

[Alter99] S. Alter. "Information Systems: A Management Perspective". New-Jersey: Addison-Wesley, 1999.

[Bañares97] R. Bañares-Alcántara, J.M.P. King. "Design Support Systems for Process Engineering-III Design Rationale as a Requirement for Effective Support". Computer and Chemical Engineering Vol. 21, No.3, pp 263-276, 1997.

[Daniels94] N.C. Daniels. "The impact of IT on the organization". In: Information Technology. The Management Challenge. New Jersey: Addison-Wesley Publishing Co, 1994, p.112-116.

[FIPA97] “Agent Communication Language: FIPA 97 Specification, Version 2.0, part 2”. Foundation for Intelligent Physical Agents, 1997.

[Hammer96] M. Hammer. “The triumph of process”. In Beyond Reengineering. Harper Bussiness, 1996, p. 3-17.

[Han95] C. Han, J. Douglas, G. Stephanopolos. “Agent-Based Approach to a Design Support System for the Synthesis of Continuous Chemical Process”. *Computer Chemical Engineering*, vol 19 Suppl., pp S63-S69 1995.

[Maturana99] F. Maturana, W. Shen, D. H. Norrie: "MetaMorph: An Adaptive Agent-Based Architecture for Intelligent Manufacturing” *Int. Journal of Production Research*, 37(10), 1999, 2159-2174.

[Odell00] J. Odell, H.Van Dyke Parunak, B. Bauer. “Extending UML for Agents”. *Submitted paper, 2000.* (see http://www.jamesodell.com).

[Parunak98] H. Van Dyke Parunak. “Practical and Industrial Applications of Agent-Based Systems”. *Report: Industrial Technology Institute*, 1998

[Prietula98] J. Prietula, K.M. Carley, L. Gasser (editors). “Simulating Organizations. Computational Models of Institutions and Groups”. Menlo Park: AAAI Press/The MIT Press , 1998.

[Pynadath99] D.V. Pynadath, M. Tambe, N. Chavaut, L. Cavedon. “Towards team-oriented Programming”. *Proceedings of Agent Theories, Architectures and Languages* (ATAL99).

[Ribeiro96] R.A. Ribeiro. “Fuzzy multiple attribute decision making: A review and new preference elicitation techniques”. *Fuzzy Sets and Systems*, vol. 78, 1996, 155-181.

[Torra98] V. Torra. “On the integration of numerical information: from the arithmetic mean to fuzzy integrals”. Research Report DEI-RR-98-010, Universitat Rovira i Virgili, 1998.

[Simeoff98] J. Simeoff J. and M. L. Maher. “Ontology-based multimedia data mining for design information retrieval”. Proceedings of ACSE Computing Congress, Cambridge, 1998.

[Struthers98] B. Struthers and R. Curwen. “A new approach to integrated process system engineering – The VIBE agent environment” *Computers & Chemical Engineering*. Vol. 22, Suppl. S745-S748, 1998

[Tambe00] M.Tambe, D. Pynadath, N.Chauvat. “Adaptive Agent Integration Architectures for Heterogeneous Team Members”. *Proceedings of the International Conference on Multi-Agent Systems (ICMAS 2000).*

[Tambe97] M. Tambe. “Towards Flexible team-work”. *Journal of Artificial Intelligence Research* 7 (1997) pp. 83-12

[Westerberg97] A.W. Westerberg, E. Subrahmanian. Y. Reich. S. Konda & the n-dim group. “Designing the Process Design Process”. *Computers & Chemical Engineering.* Vol. 21, Suppl. S1-S9, 1997