

# A METHODOLOGY FOR SHOP FLOOR REENGINEERING BASED ON MULTIAGENTS

---

José Barata , L. M. Camarinha-Matos  
New University of Lisbon  
Quinta da Torre – 2825 114 Caparica – PORTUGAL  
jab@uninova.pt -- cam@uninova.pt

*Achieving shop floor agility is a major challenge for manufacturing companies. A multiagent based approach for shop floor reengineering where agility is achieved through configurations of contracts is briefly introduced. A methodology to agentify manufacturing components in order to participate in the multiagent community is presented. An experimental validation scenario is finally described.*

## 1. INTRODUCTION

Shop floor agility is a central problem for manufacturing companies. Internal and external constraints, such as growing number of product variants and turbulent markets, are changing the way these companies operate and impose continuous adaptations or reconfigurations of their shop floors. This need for continuous shop floor changes is so important that finding a solution to this problem would offer a competitive advantage to contemporary manufacturing companies.

The central issue is, therefore, to design and develop techniques, methods, and tools are appropriate to address shop floors whose life cycles are no more static but show high level of dynamics. In other words, how to make the process of changing and adapting the shop floor faster, cost effective, and easy. The long history of industrial systems automation shows that the problem of developing and maintaining agile shop floors cannot be solved without an integrated view, accommodating the different perspectives and actors involved in the various phases of the life cycle of these systems. Moreover, supporting methods and tools should be designed and developed to accommodate the continuous evolution of the manufacturing systems along their life cycle phases – a problem of shop floor reengineering. The design and development of a methodology to address shop floor reengineering is thus an important research issue aiming to improve shop floor agility, and therefore, increasing the global competitiveness of companies.

A particularly critical element in a shop floor reengineering process is the control system. Current control/supervision systems are not agile because any shop floor

change requires programming modifications, which imply the need for qualified programmers, usually not available in manufacturing SMEs. To worsen the situation, the changes (even small changes) might affect the global system architecture, which inevitably increases the programming effort and the potential for side-effect errors. It is therefore vital to develop approaches, and new methods and tools that eliminate or reduce these problems, making the process of change (re-engineering) faster and easier, focusing on *configuration* instead of *codification*. In this context, a multiagent based system, called **Coalition Based Approach for Shop Floor Agility – CoBASA**, was created to support the reengineering process of shop floor control/supervision architectures. CoBASA uses contracts to govern the relationships between coalitions' members (manufacturing agents). The main foundations of the system architecture was described in (Barata & Camarinha-Matos, 2003). This paper focuses on the reengineering methodology to be used with CoBASA. First, the main agents that compose the CoBASA architecture are described in section 2, while section 4 describes the steps required to operate CoBASA, which involves creating Manufacturing Resource Agents - MRAs that can be used as candidates in future manufacturing coalitions. Then, section 4 briefly describes the experimental setup. Finally, section 5 presents the conclusions.

## 2. THE CoBASA ARCHITECTURE

Although the the main aspects (components and basic interactions) of the CoBASA architecture have been described in (Barata & Camarinha-Matos, 2002, 2003; Camarinha-Matos & Barata, 2001) a brief overview is presented here for the sake of better understanding of this paper.

The basic components of the proposed architecture are Manufacturing Components, Manufacturing Resource Agents, Coordinating Agents, Clusters, Coalitions/consortia, Broker, and Contracts.

### **Definition 1 - Manufacturing Components**

A manufacturing component is a physical equipment that can perform a specific function in the shop floor. It is able to execute one or more basic production actions, e.g. moving, transforming, fixing or grabbing.

### **Definition 2 - Manufacturing Resource Agents (MRA)**

The MRA is an agentified manufacturing component, i.e. a manufacturing component extended with agent skills like negotiation, contracting, and servicing, able to participate in coalitions/consortia.

As it could be expected there are several types of MRAs, one for each manufacturing component type. Therefore it is expected to find robot MRAs, gripper MRAs, tool warehouse MRAs, etc. Each MRA is individualised by its basic skills and attributes. In the CoBASA society the basic members are not the physical manufacturing components but the MRAs. Each manufacturing component thus needs to be agentified (transformed into an MRA) before it can participate in the CoBASA society. Since skills represent a very important characteristic of a

manufacturing component, and since these skills are implemented by manufacturing controllers, a MRA represents in fact the agentification of a manufacturing controller.

Every MRA should be able to: 1) adhere to a cluster, 2) participate in consortia/coalitions, and 3) perform the manufacturing operations associated to the skills it represents.

**Definition 3 – Coalition/Consortium**

A coalition/consortium is an aggregated group of agentified manufacturing components (MRAs), whose cooperation is regulated by a coalition contract, interacting in order to generate aggregated functionalities that in some cases are more complex than the simple addition of their individual capabilities.

A coalition is usually regarded as an organisational structure that gathers groups of agents cooperating to satisfy a common goal (Shehory & Kraus, 1995). On the other hand, the term consortium comes from the business area where it is defined as an association of companies for some definite purpose. Comparing both definitions it can be seen that they are quite similar because in both definitions there is the notion of a group of entities cooperating towards a common goal. Therefore, this common definition is adapted to the CoBASA context.

A basic coalition/consortium besides being composed of MRAs includes an agent that leads the coalition – Coordinating Agent (CA). In addition it can include as members other coalitions/consortia. The coordinator of a consortium is able to execute complex operations that are composed of simpler operations offered by the consortium members.

**Definition 4 – Coordinating Agent (CA)**

A CA is a pure software agent (not directly connected to any manufacturing component) specialised in coordinating the activities of a coalition, i.e. that represents a coalition.

As members of coalitions/consortia, MRAs can only play the member role while CAs can play both the coordinator role and member role. A simple manufacturing coalition/consortium is composed of some MRAs and one CA. However, a coalition/consortium can be composed of other consortia creating in this way a hierarchy of coalitions/consortia. Therefore a CA can simultaneously coordinate MRAs and others CAs.

It is worthwhile to emphasise an intuitive aspect: the fact that the set of skills offered by a coalition is composed of not only the basic skills brought in by its members but also more high level skills that result from a composition of those simpler skills. Therefore, some kind of skill composition is needed to generate new skills.

When forming a coalition/consortium there are no limitations on the type of agents that can be involved in but it is mandatory to know what are the available and willing to participate agents. It would be important that these agents could be grouped by their spatial relationships (or any other relevant relationship e.g. technological compatibility), i.e., manufacturing agents that could establish

consortia should be grouped together because they share something when they are candidates to consortia. Therefore, there is a need for a structure (cluster) that group the agentified manufacturing components (MRAs) willing/able to cooperate and from which the agents share some concepts.

**Definition 5 – Shop Floor Cluster**

A cluster is a group of agentified manufacturing components (MRAs) committed to participate in coalitions/consortia and sharing some relationships, like belonging to the same manufacturing structure and possessing technological compatibility.

A shop floor cluster includes a kind of directory where the agents willing to participate in coalitions/consortia can register. This directory acts as the place where those agents become known and where they publish their skills. A special agent – the cluster manager (CMgA) – is responsible for keeping the directory and supporting the adhesion/withdrawal of agents.

The formation of a coalition/consortium, however, is not done by the cluster manager but rather by a specialised agent called broker.

**Definition 6 – Broker**

A broker is an agent that is responsible for the creation of coalitions/consortia in interaction with an external user. The broker agent gathers information from the cluster and based on the user preferences supervises/assists the process of creating the coalition/consortium.

The broker therefore interacts with the human, the cluster, and the candidate members to the consortium. Coalitions/consortia can be created either automatically or manually. At the current stage only the manual option is considered.

Contracts are the next important CoBASA mechanism, which is used to regulate the agent's interaction with a cluster as well as its behaviour within coalitions/consortia.

**Definition 7 – Contract, according to the law (FindLaw, 2002)**

“An agreement between two or more parties that creates in each party a duty to do or not do something and a right to performance of the other's duty or a remedy for the breach of the other's duty.”

In the CoBASA architecture two types of contracts are considered: **cluster adhesion contract (CAC)**, and **multilateral consortium contract (MCC)**.

**Definition 8 – Cluster Adhesion Contract (CAC)**

This contract regulates the behaviour of the MRA when interacting with a cluster. Since the terms imposed by the cluster cannot be negotiable by the MRA the contract type is “adhesion”. The CMgA offers cluster services in exchange for services (abilities or skills) from the MRA.

The CAC includes terms such as the ontologies that must be used by the candidate, the duration of the membership, the consideration (a law term that describes what the candidate should give in turn of joining the cluster, usually the skills that the candidate is bringing to the cluster).

**Definition 8 – Multilateral Coalition/consortium Contract (MCC)**

This contract regulates the behaviour of the coalition by imposing rights and duties to the coalition members. The contract identifies all members and must be signed by them to be effective. The coalition leader (CA) is identified as well as its members. The members are entitled to a kind of award (credit) in exchange for their skills.

The important terms of this type of contract, other than the usual ones like duration, names of the members, penalties, etc., are the consideration and the individual skills that each member brings to the contract. The importance of contracts as a mechanism to create/change flexible and agile control structures (consortia) lays on the fact that the generic behaviours exhibited by generic agents are constrained by the contract that each agent has signed. This calls forth that different consortium behaviours can be achieved by just changing the terms of the consortium contract, namely the skills brought to the consortium.

The MRA was defined as a manufacturing component extended with agent skills, which corresponds to its agentification in order to be able to participate in the CoBASA society. The agentification could have been achieved by developing an agent that could simultaneously interact with the manufacturing equipment controller, and manage its CoBASA social activities (cluster joining and coalition participation) that includes, among others, the contract negotiation and composition of skills tasks. In addition, the requirements imposed by the need for interaction to be maintained with the controller (short response time, interaction protocol specificities, ...) suggest the use of a dedicated agent to interact with the controller. Therefore, as the two activities are both very demanding to be accomplished by one agent only, the adopted approach was to separate the functionalities and to have one dedicated agent to interact with the controller – Agent Machine Interface (AMI) and a generic agent specialised in the CoBASA social activities, namely, contract negotiation and skills composition – Generic Agent (GA).

### **3. THE STEPS OF THE METHODOLOGY**

This section describes the main steps required to operate CoBASA. This involves creating Manufacturing Resource Agents - MRAs that can be used as candidates in future manufacturing coalitions, and creating, changing and deleting consortia. In this paper only the part referring to the join cluster is analysed since the other parts have been already discussed in (Barata & Camarinha-Matos, 2002, 2003)

Figure 1 shows these steps by clearly indicating the most important functionalities under CoBASA. While the steps for creating, changing, and deleting consortia must be executed any time the system is changed, the join cluster functionality needs only to be executed whenever a manufacturing component needs

to be agentified. Therefore, the steps required to create a manufacturing agent are done only once in the life of a manufacturing component. Considering that in most situations the manufacturing components are reused the time spent with the agentification is affordable in comparison with its lifetime. This is an important fact since the agentification process is the most complex and time-consuming activity and, in addition, most of the used functionalities are creating, changing, and deleting consortia, which are simple and fast processes.

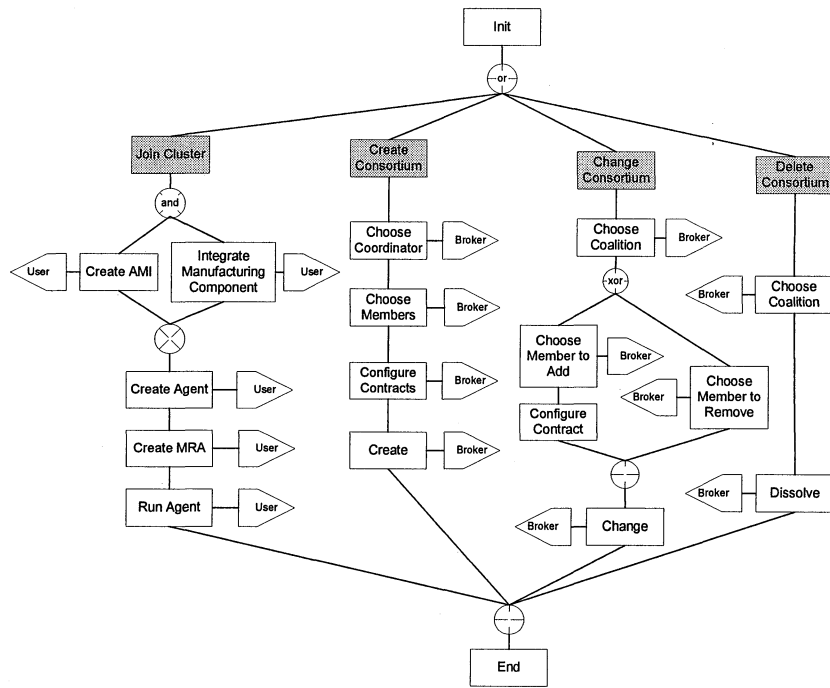


Figure 1 - Steps of the methodology

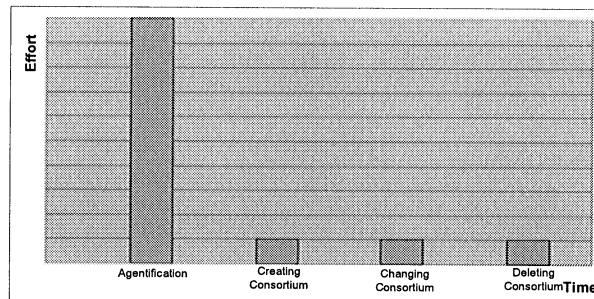


Figure 2 – Variation of the configuration effort for a manufacturing component

Figure 2 illustrates how the effort required to configure a manufacturing component evolves along its lifetime. The objective is not to indicate exact values about how much effort is required but rather to give an idea that the agentification

process that is made initially, and only once, requires a bigger effort than the other phases of the component's lifetime. In the specific case of the manufacturing component considered in Figure 2, the manufacturing agent after being agentified, participated in a consortium, which was later on changed, and finally removed.

## 2.1 Join the Cluster

The branch *Join Cluster* in Figure 1 shows the steps that must be performed in order to have a manufacturing agent participating in a cluster. Therefore, before joining the cluster the manufacturing equipment must be transformed into a manufacturing agent (MRA). Joining the cluster corresponds to the practical situation in which a manufacturing component is added to a given manufacturing cell. It is supposed that the broker agent and the cluster manager agent are already running. It must be noticed that a MRA is more than an agentified manufacturing component and thus the following steps have been identified whenever a new manufacturing equipment needs to be transformed into a MRA: 1) *Create the AMI*, 2) *Integrate Manufacturing Component*, 3) *Create Manufacturing Agent – Agentification of the Manufacturing Component*, 4) *Create the MRA*, and 5) *Run the agent*.

Steps 1 to 3 correspond to the agentification of the manufacturing component. At the end of step 4 a manufacturing resource agent exists composed of the configured generic agent plus the appropriate AMI for the manufacturing component. At the end of step 5 the agent has been registered in the cluster.

**Create the AMI.** In this step the user configures the generic AMI according to the functionalities and requirements of the manufacturing component to which the AMI is going to be connected. The AMI establishes the link between the generic agent (GA) and the manufacturing component. This agent is specific to the type of the manufacturing component to which it is connected in terms of the functionalities it offers.

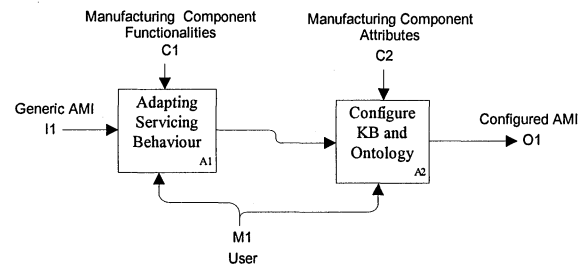


Figure 3 - Activities when configuring the AMI

The generic AMI includes roughly the behaviours required for user interface, and the generic behaviour to attend requests from the GA to which it will be connected. Configuring it corresponds to implement the specific functionalities of the agent to which it will be connected. This is represented in Figure 3 by the activity *Adapting Servicing Behaviour*. Furthermore, it is necessary to configure the individual KB and ontology of the agent to the individual attributes of the

manufacturing component, which is represented by the activity *Configure KB and Ontology* in Figure 3.

**Integrate Manufacturing Component.** In this phase the functionalities of the manufacturing component controller are modelled using computer based abstraction mechanisms such as Remote Procedure Calls (RPCs) or distributed objects. When using RPCs, the functionalities of the controller are represented by the methods included in the RPC that models the controller. When modelling manufacturing controllers using objects or distributed objects, their attributes model the static characteristics of the controller being modelled while the methods model the controller's functionalities. This modelling is fundamental since many of today's manufacturing controllers do not provide an abstraction at this level and, hence, to be controlled, they need special commands usually sent via a RS232 protocol. This is the process of connecting the physical controller to the agent. This could be an easy task if every physical component was controlled directly by its own agent. However, outdated legacy controllers with closed architectures control most of the existing physical components.

Independently of how the commands are sent to the specific manufacturing controller the important thing to keep in mind is that it is necessary to adapt the computational abstraction of the manufacturing component to an abstraction level that can be used by agents. This is so because, in the end, the commands that the physical manufacturing controllers receive are originated in the agents that compose CoBASA (MRA agents).

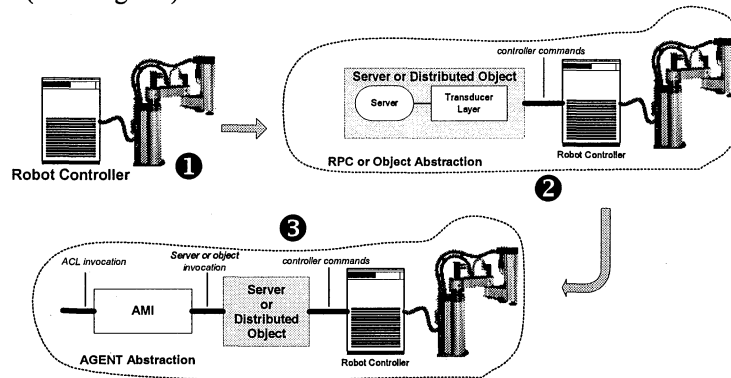


Figure 4 - Transformation of a manufacturing component into an agent

The objective of this phase is thus to build a server, using RPCs or a distributed object, using CORBA or DCOM, whose methods can be invoked from any computer system connected to a network. This approach decouples the physical manufacturing controller from the agent environment, which facilitates the integration. To integrate these legacy components in the agents' framework it is thus necessary to develop a software wrapper to hide the details of each component (Barata & Camarinha-Matos, 1995; Camarinha-Matos, Seabra Lopes, & Barata, 1996). The wrapper acts as an abstract machine to the agent supplying primitives that represent the functionality of the physical component and its local controller. The steps of this phase are indicated in Figure 4 with the numbers 1 and 2. At the



end of this phase the manufacturing component can be commanded through the activation of a RPC or the invocation of one of the methods of the object that mimics it.

**Agentification of the Manufacturing Component.** The physical manufacturing component is effectively transformed into a manufacturing agent at the end of this phase, hence, the name agentification. This is represented in the number ③ of Figure 4.

Steps 1 to 3 occur only when the physical manufacturing component is first integrated in the community of agents. All the other situations happen after the component has been already agentified, like for instance changes on the consortium.

The activity that must be done in this phase is connecting the AMI, which has been already configured in step 1, to the object or server that represents the physical manufacturing controller. The AMI accesses the wrapper services using a local software interface (proxy), where all services implemented by the wrapper/legacy controller are defined. Figure 6 illustrates how the AMI connects to the wrapper by showing some details of it that were not presented in Figure 4.

The generic AMI agent is a simple agent with a simple behaviour to accept requests from other agents. When a REQUEST is received, the AMI calls the wrapper to execute the requested service and when the command is executed, it sends back a DONE message to the enquirer agent. Each AMI implements the services supported by the physical component by configuring what the services of the proxy to which it is connected are and the name/address of the component. Because issuing Agent Communication Language REQUEST commands to the AMI does the activation of the manufacturing component, it can be stated, then, that this software layer provides agent abstraction (Figure 4, number ③).

**Create the MRA.** At the end of this phase a Manufacturing Resource Agent (MRA) is created. MRAs are composed of a Generic Agent connected to the agentified manufacturing component produced in the last phase. In the CoBASA framework only manufacturing components represented in this way (MRAs) are able to join the cluster and, hence, participate in coalitions.

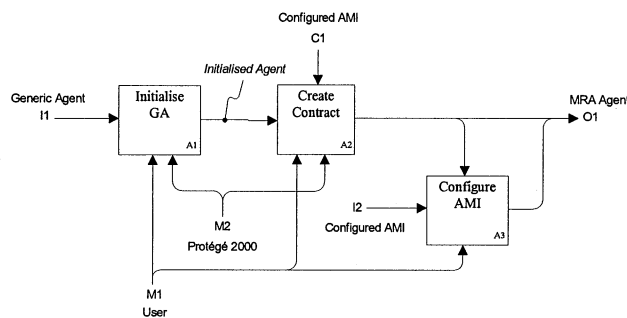


Figure 5 - Activities in the Create MRA phase

In Figure 5 the activities that must be executed to create a MRA are shown. In the activity *Initialise GA* the user initialises the Generic Agent, which, in this case, is

used to supply negotiation skills to the MRA agent. It is then necessary to configure the initialisation and the ontology file to be used by the agent. At the end of the activity, the GA is an initialised instance of a generic agent.

The connection to the AMI is guaranteed by creating a consortium contract between the generic agent and the AMI, establishing in this way a consortium. The member promise part (AMI) of the contract contains the services supplied by the AMI. This is supported by the activity *Create Contract*, which is done by the user using the Protégé 2000 ontology management (Protégé-2000, 2000) environment to create the contract. The information existing in the *configured AMI*, which was configured before, constrains the output of this activity because the behaviour of the MRA agent depends on the skills brought in by the AMI to whom it is connected. The contract being created is attached to the variable *coordinated contracts* that contains all the contracts that this GA is coordinating. In this situation, the GA only coordinates, in fact, the AMI to which it is connected.

At the end of the *Create Contract* activity the MRA agent is complete and composed of the generic agent GA attached through *coordinated contracts* to its AMI. However, the AMI is not yet configured in the sense that it should only accept requests from the GA that is attached to it. This is done in the activity *Configure AMI* in which the AMI must also be configured to include the name of the generic agent to which it is connected. This guarantees that an AMI refuses any requests from unknown agents.

Figure 6 shows the various entities that compose the MRA. It must be remembered again that a MRA needs only to be created when the manufacturing component is used for the first time. Future uses of the manufacturing component through its representative (the MRA), in coalitions and possible different clusters, do not involve any changes to it.

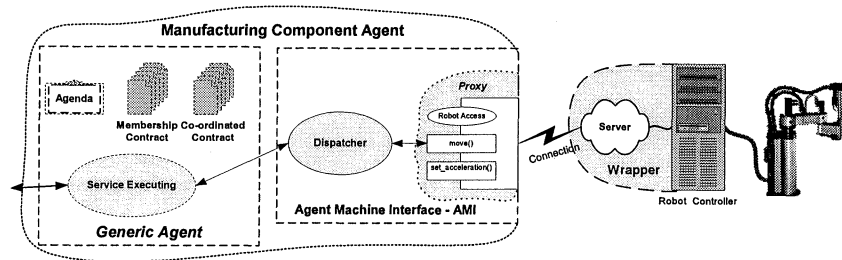


Figure 6 - The Manufacturing Resource Agent – MRA

**Run the agent.** This step is just to guarantee that the agent joins the cluster.

#### 4. EXPERIMENTAL SETUP

The described methodology was validated against a scenario that could represent a real manufacturing environment, which is the case with the flexible manufacturing system (Novaflex), established at the UNINOVA (Figure 7).

The cluster is the Novaflex which aggregates all the manufacturing equipment. Different coalitions can be created out of this equipment. These coalitions represent no more than different ways of exploring the cell. Therefore, there are two ways of

regarding the cell: the physical and the abstract. In the physical way the NovaFlex is a cell composed of several manufacturing equipments that are related by physical relations. It is possible to imagine that parts of the entire cell can be operated independently as well as that equipment can be added or removed. In the abstracted way the NovaFlex is a cluster composed of several manufacturing agents (agentified manufacturing equipment) whose entire set of skills represents the potential of this cluster to solve problems. Whenever a problem requiring a specific set of skills available in the cluster is needed a coalition can be created fitted with that specific set of skills. Furthermore, several problems can be answered simultaneously as long as the cluster includes members able to answer the various problem requirements. The interesting point about this vision is the dynamics of the coalitions.



Figure 7 – Partial view of Novaflex

It is important to remember that any manufacturing equipment that might need to be added to the NovaFlex physical infrastructure (physical view) must join the cluster NovaFlex (abstract view). Hence, adding equipment corresponds to joining the cluster while removing equipment corresponds to leaving the cluster.

The scenario used to test the prototype included a BOSCH SCARA based assembly system as well as an ABB based assembly system.

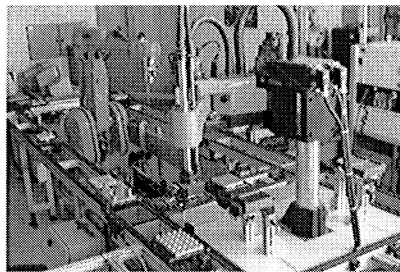


Figure 8 - The SCARA assembly system

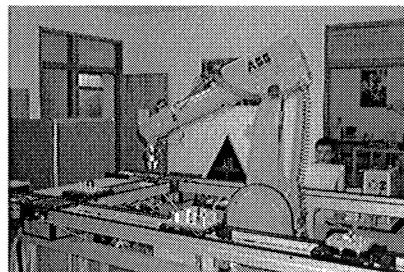


Figure 9 - The ABB assembly system

The SCARA assembly system (Figure 8) is composed of one robot BOSCH SCARA SR-80 equipped with a tool exchange mechanism, a tool warehouse composed of 6 individual slots, 4 different grippers, a feeder, and a fixture.

The ABB assembly system (Figure 9) is composed of one robot ABB IRB 2000 equipped with a tool exchange mechanism, a tool warehouse composed of 4 individual slots, 4 different grippers, a feeder, and a fixture.

Two coalitions were successfully formed out of each of these two subsystems.

## 5. CONCLUSIONS

This paper described a methodology to support the agility and reconfiguration of shop floor control systems. The CoBASA multiagent architecture that supports the shop floor reengineering process was briefly described and the methodology used to create the agentified manufacturing components was detailed as well as the experimental setup where the prototype was applied.

The first results of using this reengineering methodology proved the validity of the concept in terms of agility. Changes made to the manufacturing cell, either by adding or removing manufacturing components, were simple and involved only changes in the contracts that regulate the created coalitions. In addition the creation of different ways of exploiting the manufacturing cell (NovaFlex) was simple since it involved only the creation of new coalitions.

## 6. ACKNOWLEDGEMENT

This work was funded in part by the European Commission through the ASSEMBLY-NET Growth Network.

## 7. REFERENCES

1. Barata, J., & Camarinha-Matos, L. M. (1995). Dynamic Behaviour Objects in Modelling Manufacturing Processes. In Q. Sun & Z. Tang & Y. Zhang (Eds.), *Computer Applications in Production Engineering* (Vol. 1, pp. 499-508). London: Chapman & Hall.
2. Barata, J., & Camarinha-Matos, L. M. (2002). *Implementing a Contract-based Multi-Agent Approach for Shop Floor Agility*. Paper presented at the Holomas 2002, Aix-en-Provence.
3. Barata, J., & Camarinha-Matos, L. M. (2003). Coalitions of Manufacturing Components for Shop Floor Agility - The CoBaSA Architecture. *International Journal of Networking and Virtual Organisations*, 2(1), 50-77.
4. Camarinha-Matos, L. M., & Barata, J. (2001). Contract-Based Approach for Shop-Floor Reengineering. In R. Bernhardt & H. H. Erbe (Eds.), *Cost Oriented Automation* (First ed., Vol. 1, pp. 141-148). Oxford - UK: Pergamon - Elsevier.
5. Camarinha-Matos, L. M., Seabra Lopes, L., & Barata, J. (1996). Integration and Learning in Supervision of Flexible Assembly Systems. *IEEE Transactions on Robotics and Automation (Special Issue on Assembly and Task Planning)*, 12(2), 202-219.
6. FindLaw. (2002). *Legal Dictionary* [Web Site]. Retrieved November 2 002, 2 002, from the World Wide Web: <http://dictionary.lp.findlaw.com/>
7. Protégé-2000. (2000). <http://protege.stanford.edu> [web site]. Retrieved Jan 2002, 2002, from the World Wide Web:
8. Shehory, O., & Kraus, S. (1995). Coalition Formation among Autonomous Agents: Strategies and Complexity. In C. Castelfranchi & J. P. Muller (Eds.), *From Reaction to Cognition* (pp. 57-72). Heidelberg: Springer-Verlag.