Towards Agile Scheduling in Extended Enterprise

R. J. Rabelo [a] ; *L.M. Camarinha-Matos* [b]
[a] *Federal University of Santa Catarina / Brazil, at New University of Lisbon / UNINOVA - Portugal (e-mail : kadu@uninova.pt)*
[b] *New University of Lisbon and UNINOVA - 2825 Monte da Caparica - Portugal (e-mail : cam@uninova.pt)*

Abstract
This paper presents the UNL's approach to extend the HOLOS agile scheduling system towards the extended enterprise. The concept of extended enterprise and related problems arc brought up, as well as the notion of agile scheduling. The HOLOS system architecture is briefly explained. The necessity to develop a high level decision support system (DSS) to manage the scheduling of a business process during its production in the network of enterprises is emphasized, as well as its main functions are presented. A general framework to integrate the DSS with the scheduling system is also shown. Finally, a flexible communication infrastructure to support the integration of an enterprise in the network is introduced.

Keywords
Agile Scheduling, Extended Enterprise, Distributed Business Process, Multiagent Systems

1 INTRODUCTION AND MOTIVATION

The application of the extended or virtual enterprise (EE) paradigm (Browne,1995) by the industries starts to be a strong trend nowadays. Due to the increasing of competitiveness and market globalization, companies look for a more tight control and, at the same time, a more amicable relationship with their suppliers and clients. In an EE scenario, the production of an entire product (a business process - BP) can be performed by many different and geographically distant suppliers, connected to each other via an electronic network. In a certain moment, for a certain product, a given enterprise (a 'node' of the network) acts as the *prime* one (the client) and the others as its *servers*. However, in order to achieve the necessary agility, this (electronic) production chain has to be efficiently coordinated so that any production problem (in any node and which can affect the whole "production graph") can be quickly detected, and hence adequate decision can be made. Therefore, the necessity for a high level scheduling activity, which involves logistics aspects, ascends in significance. It docs not mean the other activities like planning and execution supervision are not strongly affected too. However, this paper will focus on scheduling only.

Nevertheless, the development of a scheduling system that can be able to face such necessity within an EE scenario is still a challenge. Beyond the complexity of this high level coordination itself, the new required functionalities the scheduling systems shall have to deal with are not precisely defined yet. Moreover, the more suitable technologies which can be applied on information integration and communication infrastructures are not completely understood. In fact, in spite of the running international research projects in the EE domain (such as

L. M. Camarinha-Matos et al. (eds.), *Balanced Automation Systems II*

(NIIP,1995)) as well as some commercial software products already available (Cabletron,1995), we may say they are still in phase of maturation. On the other hand, the need for applications on EE area tends to increase highly. According to some forecasts, enterprises will invest up to 45% of their budget for information technology (IT) in EE applications by the year 2002. In fact, many enterprises have already sought for more flexible scheduling systems, with greater reaction capabilities.

A scheduling system has, as its basic function, the assignment of industrial orders (jobs) to production resources along the time, according to the production objectives and constraints. We call an Agile Scheduling System one which supports :

- Dynamic scheduling : dynamic adaptation of the current schedule in the presence of unexpected events, both from the shop floor (like machine failure, tool unavailability and operation lateness) and from the planning level (like changes in order priority and orders cancellation), and;
- Agile adaptation : flexible adaptation of the whole production structure according to a given business process' requirements. It is related to flexibility in terms of internal routing as well as production areas, or even to comprise a virtual manufacturing (Hitchcock,1994).

Under a classical scenario, an enterprise usually plans itself and runs 'isolated' from its suppliers, with a very low degree of cooperation with them and vice-versa. Enlarging the definition of agile scheduling towards EE means giving the enterprises conditions to run under a truly on-line just-in-time system. In fact, the application of the EE paradigm brings the possibility to manipulate and integrate all necessary information since it flows (on-line) through the network and then it is possible to treat them computationally. The issue of conventional means, like telex, telephone, telefax and even the electronic mail, tends to decrease considerably, given that most of the services for which these ways are necessaries will be provided in the form of network services. Therefore, it is not a simple question of making the enterprises to talk to each other about their businesses, but also to electronically synchronize their production for a certain BP. This synchronization is a tough task. An entire BP is constituted by several interdependent (sub)BPs, which are produced by several enterprises. Hence, problems like BP delay, BP cancellation / modification and changes on BP priorities, becomes much more difficult to be managed. Any decision upon a BP which affects the production channel cannot be decided alone, but by means of a conversation between all the enterprises implicated in it. Such conversation can involve an analysis, by each enterprise, in terms of strategies and decisions to be taken accordingly, which appears to be relatively complex. Therefore, a **decision support system** to evaluate and help conflicts resolution related to those problems as well as to supervise the entire BP scheduling becomes necessary.

This paper presents the UNL's approach to extend its works on agile scheduling towards EE. Firstly, the general architecture of our scheduling system (the HOLOS system architecture) is presented. Secondly, an integration infrastructure to support both the enterprise networking and agile scheduling is shown. Thirdly, our approach to the envisaged decision support system is introduced. Finally, some remarks and open questions are pointed out. This work uses a balanced automation philosophy at various levels. The integration infrastructure architecture allows a flexible and gradual enterprise integration in the network and, at the same time, guarantees the enterprise independence and privacy. It applies the legacy system approach in integrating all existing systems into the global architecture. The human experience and flexibility is exploited in the design and DSS operation.

2 THE HOLOS ARCHITECTURE

The HOLOS architecture was designed to cope with the intended mentioned agility (Rabelo,1995). It is supported by three elements : a) Modeling and Information Integration; b) Virtualization of the enterprise's production structure; and c) Multiagent Distributed Control.

2.1 Modeling and Information Integration

A CIM Information System (CIM-IS) (Osório,1993) is used to integrate CIM activities and associated systems. We assume the CIM-IS as the source of all information required for scheduling, such as process plans, models of production resources and production plans. The frame-based / object-oriented modeling paradigm is applied in all components of the architecture for information modeling and knowledge representation, and include some results achieved in data modeling in various international projects.

A task to be executed is modeled in the form of "business processes" (BP), "enterprise activities" (EA) and "procedural rule sets" (PRS), following the CIM-OSA methodology (AMICE,1993) to model the enterprise's dynamic processes. A BP corresponds to an abstract task description (a production order in our case) which can be decomposed either into a set of EAs or BPs. An EA represents an installed functionality; i.e. the operation description, which cannot be further decomposed. It may correspond to an individual process plan operation. The PRS represents the dynamic control flow between BPs and/or EAs, that is, it specifies the precedence relationships and their enabling conditions. A production order is therefore viewed as a logical, temporary and hierarchical grouping of these concepts, and it represents how an entire BP must be accomplished. In fact, this grouping becomes a tree - *a BP-tree* - in which the EAs are the leaves, and BPs are the intermediate nodes. Once planned BPs (and their respective decompositions) 'arrive' they have to be scheduled according to the PRSs. The HOLOS scheduling system considers the production plan under a BP-tree form.

2.2 Virtualization of the Enterprise's production structure

A virtualization of the production resources' structure aims at a balanced distribution of business processes and a more rational use of the production resources, taking into account not only a particular area but the enterprise as a whole. In this sense, a situation of overloading or idleness upon some production resources / production areas during a scheduling may be reduced. The intended scenario is illustrated in Figure 1 : all enterprise's production structure has to flexibly adapt itself in order to both attend business processes arrivals and to react in the presence of disturbances; in other words, it has to support an agile manufacturing. The virtualization levels shown correspond to a multilevel modeling of the production structure.

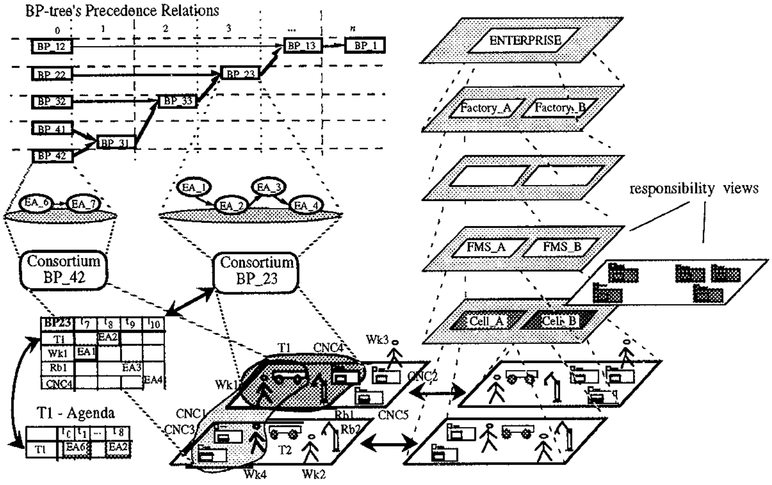

Figure 1 The intended agile scenario.

At physical level the production resources (illustrated as workers (Wk1,...,Wk4), CNC machines (CNC1,...,CNC5), transportation systems (T1 and T2), and robots (Rb1 and Rb2)) are modeled in a multilevel way, according to the physical shops' layout. This model can take into account existing organizations, such as workstations, cells, FMS, etc., which in turn can belong to different shops, departments, factories, and so on.

A second virtualization level is the first abstraction level of the production resources. It corresponds to a *responsibility* view (AMICE,1993) - a clustering - of the shop. Different clustering criteria can be applied in different shops, according to the topological relationships, and per functionality for instance. The *Consortium* is an intelligent abstraction level of the enterprise's production resources, which links the organizational and functional structure of the enterprise. A Consortium is a logical and temporary clustering of production resources dynamically joined to perform a given BP (and its EAs).

2.3 Multiagent Distributed Control

The use of a Multi-Agent Systems (MAS) paradigm (Huhns,1987) in HOLOS framework means that a scheduling is made and executed by a set of distributed cooperating agents. It aims at reaching the desired reaction capacity within the enterprise via a complementary and cooperative relationship between the agents involved in the scheduling. However, it is a awfully arduous state to achieve. The manufacturing environment is dynamic, normally over-constrained and unpredictable to some extent. Besides that, current constraints are commonly conflicting to each other so that a trade-off or requirements relaxation has to be *negotiated.* A Contract-net protocol / Negotiation paradigm (Davis,1983) has been applied to support the desired flexibility in conflicts resolution during a schedule generation and execution. The Negotiation process in HOLOS approach for scheduling consists of a process that leads agents to exchange information with other agents about a given BP/EA's requirements until a production-resource-agent (robots, CNC machines, etc.) is selected to execute it.

The HOLOS agents

A HOLOS agent can be defined as a cooperative and semi-autonomous software module with : a) some intelligence and knowledge on how to execute a BP/EA; b) some knowledge about the other agents; and c) the capability to communicate with the other agents.

A HOLOS agent is modeled as an object and represented by a frame. The HOLOS agents are heterogeneous, functionally hierarchical, semi-autonomous, deliberative, and they are designed according to the four problem solving phases in Distributed Artificial Intelligence : problem decomposition, sub-problem distribution, sub-problem solution and answer synthesis (Qing,1988). There are four "classes" of HOLOS agents :

- Scheduling Supervisor agent (SS) - It is a semi-hierarchical supervisor agent, responsible for the global scheduling supervision. SS as basic functions : loading both BP-trees and involved information models from the CIM-IS; definition of EA requirements and their sending to the Local Spreading Centers agents; creation of Consortia; high level changes on planned BP/EAs; high level actions for conflicts resolution; and visualization and scheduling evaluation. It is the representant of the scheduling system within the manufacturing environment.

- Local Spreading Center agent (LSC) - It represents a decentralized control structure for EA requests spreading. Its main function refers to negotiate with the production resources agents in order to select one for the execution of each EA. Due to its responsibility view, a LSC knows who are the adequate production resource agents which cope with the EA's requirements. Thus, since an EA request is received, the LSC re-sends it to all "registered" and operational production resource agents. At the end of such a EA spreading, the LSC has to evaluate the proposals come from the production resource agents in order to select the *best* one based on some criteria, specified into the LSC's knowledge.

- Enterprise Activity Agent (EAA) - It represents a virtualization of the production resource's local controller. From the organizational point of view it is associated to the Physical Level.

From the scheduling control point of view, the EAA has, as essential functions, "self-supervision", receiving EAs, making their evaluation, and further answering to a LSC about its either positive or negative acceptance. The EAA architecture is composed by two basic interacting processes, a Manager and a Server - tandem architecture. The Server represents the resource's local controller which gives to its Manager an allowance for offering services it is capable to execute. The Manager "represents" this Server within the manufacturing environment. Its basic function is "to sell" (via negotiation) the Server's services. The Manager centralizes all decisions and communications related to the Server it represents. In spite of an EAA being not a single process, the architecture sees it as a single entity.

- Consortium agent - It represents the highest virtualization level of EAAs. From the organizational point of view it is associated to the Dynamic Virtual Level. The objective is to select a team - *logically* organized - of EAAs to execute a BP instead of assigning it necessarily to a pre-selected team - *fixed* organization. A Consortium is a dynamic, temporary and logical clustering of EAAs planned (selected) to execute a whole BP. Therefore, there will be as many Consortia as planned BPs to be executed. Since each Consortium has its local / particular schedule, we may say the scheduling is decentralized. An EAA can belong to several Consortia along the time. Since an EAA finishes the execution of contracted EA(s) for some Consortium, the EAA becomes free both to execute other EA(s) already contracted with another Consortium and to look for more EAs, which are still waiting for execution proposals. At the end of an entire BP execution the Consortium agent kills itself. A Consortium agent has three basic control functions : a) to communicate with the SS agent about the execution status of its EAs / BP; b) to supervise the EAA(s) assigned to it; c) to allow a schedule evaluation. During the life cycle of a Consortium, a substitute for an EAA has to be found - via negotiation - if it becomes non-operational, that is, a rescheduling has to be made and so a new negotiation process has to be triggered out. A Consortium agent tries to solve its local problem in an autonomous manner so that the execution of other Consortia is potentially not disturbed. However, in some cases, a Consortium cannot solve all its conflicts. In this situation the control is passed on to the SS agent, which in turn verifies which BPs (and their EAs) have to be also rescheduled.

Therefore, in the HOLOS architecture, a scheduling system is a distributed collection of instances of HOLOS agents classes, organized for a concrete structure and configuration of production resources (machines, robots, transport systems, cells, etc.), and that can perform a schedule (supported by Negotiation) and that can be supervised during its execution.

The prototype developed for the UNINOVA's FAS/FMS pilot system (NOVAFlex) is implemented in a PC 486, with the Linux operating system, and the agents are implemented in C and Prolog (Rabelo,1996). The graphical interfaces are implemented in TCL/TK. The communication between the agents and other entities of the architecture is made using standard communication protocols. The SDAI (STEP Data Access Interface) (Fowler,1992) is the standard protocol used by the agents to have access to the CIM-IS, and it is supported by RPCs. MMS (Mackiewicz,1994) is (partially) used as the protocol to support the communication with the NOVAFlex's production resources. The HOLOS agents communicate to each other using a Negotiation protocol (based on (KQML)) and TCP/IP.

3 INTEGRATION INFRASTRUCTURE FOR EXTENDED ENTERPRISE

An important requirement for the realization of the EE concept is an infrastructure that provides an open platform to support adequate IT protocols and mechanisms so that heterogeneous enterprises can inter-operate with several value chain networks. This infrastructure has to support a set of functionalities, which comprises the information exchanges among different enterprises, the update of joint information in another enterprise, and the broadcast of information that is of interest to all enterprises in the network. Both Internet and private networks can be considered.

In a real scenario of a network of enterprises, there might be thousands of products (BPs) which have to performed, and so thousands of virtual production chains in simultaneous, one for each BP. A virtual production chain is dynamically created and its life is temporary. The virtual composition (connection) ends when the entire BP production is finished. The production of a certain BP is coordinated by the respective client-enterprise (which has contracted the other enterprises). Depending on the BP (i.e. the product model), an enterprise can play the role in the respective virtual production chain as a :

- *supplier* : when it supplies a (sub)BP to the intermediate-enterprise or directly to the client-enterprise;
- *intermediate* : when it receives a (sub)BP from its suppliers, makes some transformation on it, and supplies to the client-enterprise;
- *client* : when it receives a (sub)BP from the supplier-enterprise and/or from the intermediate-enterprise, and makes the final transformation (i.e., the end product).

A traditional network of enterprises is composed by one large enterprise and SMEs around it, and normally, strongly dependent from it. While the first one concentrates its attention on the core product, its suppliers get responsible for the production of 'secondary' components. To avoid the risks that SMEs face in this scenario, a flexible infrastructure that allows them to belong to various networks and to dynamically join or quit a network is necessary. The classification below assumes a more wide and flexible EE scenario, within which the differentiation among the enterprises is expressed in terms of services they offer and hence their role in a virtual production chain, and not according to their dimension.

These enterprise 'behaviors' can occur simultaneously for the same BP. For example, an enterprise can act as a *supplier* for a certain BP's (sub)BP, and as an *intermediate* for other BP's (sub)BP. Therefore, there are two types of relationships between the enterprises involved with the production of a certain BP : horizontal and vertical. The horizontal relationship happens when a supplier-enterprise provides services/goods to an intermediate-enterprise, transparently to the client-enterprise. The vertical relationship happens when a supplier-enterprise and/or intermediate-enterprise immediately supplies the client-enterprise. We point the proposed high level scheduling coordination cannot comprise the horizontal relationships, since they are out of the direct control of the client-enterprise.

In order to preserve the enterprises privacy and security, a node has two modules for supporting its participation in the network (see figure 3) (Prodnet,1995). These two modules approach represents a strategy for dealing with legacy systems. Is spite of the necessity for the exchange of information, there is a guarantee that existing systems may run independently from the network. Namely, although there will be cooperative information exchange within the network, each node can keep its privacy and independence :

- *internal module* : represents the autonomous unit of a particular company (its PPC system). It includes the complete structure of the company's information (data bases, information systems, etc.) and all the internal decision making processes / enterprise activities.
- *cooperation module* : contains all functionalities for the inter-connection between the company and the role net. These functionalities shall involve support for product data exchange, quality control information, production information, contract negotiation, sales and purchase, etc.. Besides that, this module is also responsible for dealing with public and private information access management. It represents the communication role and works as the interlocutor of the company within the net. In terms of decision making within the company, the cooperation module is, in principle, a 'passive' entity. In short this module functions as a buffer of information input / output between the enterprise and the network. Assuming that information will flow in standard formats (like STEP, EDI, etc.), it has to be further converted into the internal formats of the particular company (internal module).

Integration of the scheduling system

The integration of the HOLOS scheduling system with the network can be made indirectly from each enterprise side (see figure 3). As already mentioned, a CIM-IS is the source of all information to the scheduling. The Scheduling Supervisor agent, as the representant of the scheduling system, takes information from the CIM-IS (which in turn also receives information from the cooperation module), according to the planned production plans, independently of the enterprise role in the network, i.e. independently of the planned BPs being related or not to other enterprises.

Even considering that the impacts and requirements of a full implantation of such infrastructure are not completely known yet, the integration of the scheduling system has not been hard. Given the flexibility and open characteristics of the HOLOS architecture, its integration is performed just by adding some adequate functionalities to the Scheduling Supervisor agent, as well as some small information complementation in the BP model.

4 A DECISION SUPPORT SYSTEM FOR GLOBAL SCHEDULING

This chapter introduces a general architecture and framework for a decision support system (DSS) to assist an enterprise in a high level scheduling. This assistance means deals with the control and synchronization of the distributed production, or in other words, to couple with a 'distributed workload management'.

Similarly to the traditional enterprise, the extended enterprise must also deal with orders requests. In this case, these orders are named as **distributed business processes** (DBP). A DBP is a dynamic and temporary set of business processes (BP) which jointly gives rise to the end-product of the extended enterprise. Figure 2 illustrates a DBP concept via its production graph. The enterprise '2' (node 2) is currently the client-enterprise, whereas node-1, node-3 and node-5 are its direct suppliers. Each node has some (sub-)BPs under its responsibility (and so the respective Enterprise Activities -EAs), placed into the dashed circle around them. They represent the value added by each enterprise on the production chain, along the time. For example, node-1 has to perform four interdependent BPs (BP1_10, BP1_22, BP1_22A and BP1_22B), whose result has to be sent to node-2 so that the BP1_1 can be done and so the whole production can be kept.

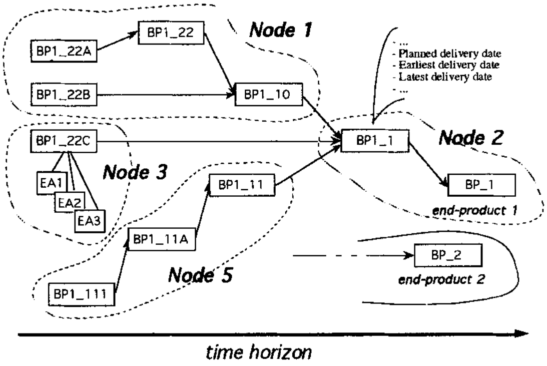

Figure 2 An example of a DBP production graph.

Since these BPs are supposed to be performed by several enterprises, the client-enterprise must manage their execution. As such, it is also liable to unexpected events, like BP delay, BP or DBP cancellation/modification, changes on DPB priorities, as well as to local communication deficiency and network overloading or failure. When one of these problems occur and the responsible enterprise cannot solve the problem locally, it offends the DBP contract, and then causes a conflict. This conflict basically influences the DBP production dates (the planned delivery date) and can affect the whole production chain. Because the components of a DBP are all inter-related, an enterprise cannot (re)plan itself for its own benefits only, but it must also look for the benefits of the whole net. Consequently, a close cooperation has to exist both intra-enterprise and inter-enterprises in order to minimize the global loss in the DBP contract * .

The complexity of the resolution of a conflict may vary quite a lot. Because an enterprise usually plans with some temporal intervals between the BPs (just to absorb the unexpected occurrences), a replanning negotiation may be relegated to adjust the planned delivery date either to the earliest delivery date or the latest delivery date. This policy could be one solution for a rapid contract re-negotiation and so for a fast DBP plan re-establishment. However, depending on the problem severity, the solution may involve a deeper - and possibly complex - analysis. Many considerations and evaluations have to be taken into account, specially regarding that an enterprise usually does not have just one DBP contracted, but plenty of them, which in turn can be indirectly affected by the problematic BP. Thus, the whole complexity makes almost impossible an user or a system solve the problem individually. Neither a single user has all knowledge and enough capabilities (in terms of time and technical background) nor a system has the natural human 'business flexibility' and experience for good trade-off analysis / decisions. In this sense, a balanced approach seems to be the suitable solution to contemplate the enterprise with these two 'knowledge sources' simultaneously. In other words, a DSS which can help and assist the user in a decision-making. In this context, a DSS has to cope with the following stages (O'Neill,1995) : identify the problem situation; data acquisition and analysis; determine causes of problem; define objectives; generate alternative solutions; comparing and evaluating alternative solutions; select (the 'best') solution.

A framework for this DSS must provide the decision-maker with selected information, describing both the competitive external environment and the way the enterprise is operating. Besides that, it must also provide well established decision-making support models and techniques. Figure 3 illustrates the general integration framework and the DSS placement, as well as the architecture relationship between the DSS with the scheduling system. Each enterprise connected to the network has two information modules, the cooperation and internal modules. The DSS is seen as another cooperation module's sub-system and can interact or have access to the other sub-systems' knowledge. The link between the DSS and the scheduling system is made indirectly, via the CIM-IS and the Scheduling Supervisor agent, the representant of the HOLOS system within the manufacturing environment.

In our current research to extend the scheduling system for dealing with the EE concept requirements as well as to develop a DSS to support a distributed workload management, several ideas from the HOLOS experience have been taken into account. From the information modeling point of view, the object-oriented paradigm / frame-based has been a very powerful tool, considering the modularity, knowledge representation facilities and re-use capabilities. However, it is necessary to evaluate the complete EE information requirements in terms of information representation and its impacts on some standards, like STEP and EDI. Likewise in the HOLOS scheduling system, a DBP monitoring / visualization in the DSS can be made by means of a Gantt Chart. We have adopted some standards for communication in the various enterprise levels (TCP/IP, RPCs, SDAI, MMS, etc.) and a CIM-IS approach. However, some complementation will be necessary in the high level protocols in order to provide the required information to the DSS. The Multi-Agent Systems approach has been useful for dealing with the integration, in a same framework, of heterogeneous sub-systems (developed in variable programming paradigms), as well as to support decentralized decision-making. The Negotiation technique has provided a good flexibility to the DSS in order to manage the disturbances in the production of a BP. However, it is still not clear, for example, the effects an on-line negotiation can provoke in the network performance. Given that the EE paradigm consequences are not completely understood and so the EE requirements, the cooperation front-end between an

* In fact, the behavior of the enterprises in a network seems to be tipically non-cooperative (self-interested) since each one tries to act according to its own objectives, which can even be a cooperative action for instance. The problem of distributed scheduling among self-interested agents (enterprises) is too complex and it is under research (Sandholm,1995). Therefore, for the current purpose of our research, we assume the phase of contract negotiation and 'business game' is passed and so the problem is related to supervise the distributed production of a DBP already contracted and which the involved enterprises will try to carry out the agreement.

automatic and human negotiation is not precisely defined yet. Moreover, the definition of all the information the DSS needs as well its knowledge base will depend on the research advances in the EE area to some extent. Finally, the EE paradigm will possibly brings up the problem of negotiation with incomplete information, a problem to be further researched.

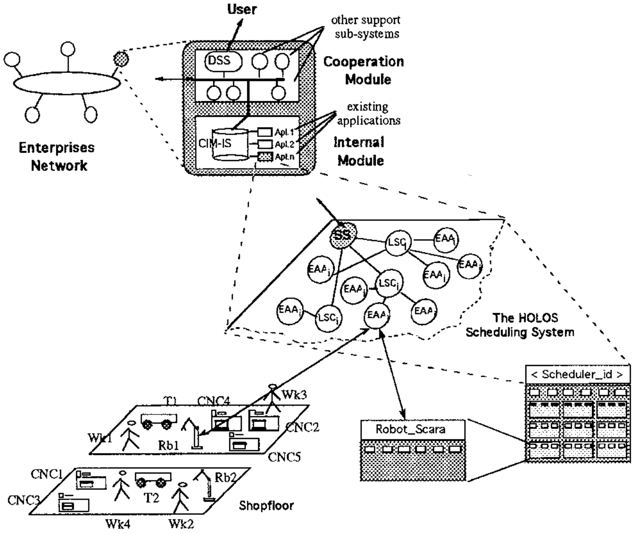

Figure 5 General integration framework.

5 CONCLUSIONS

This paper presented our current research to extend the HOLOS agile scheduling system for dealing with the EE concept requirements as well as to develop a DSS to support a 'distributed workload management'. The EE scenario was characterized and the associated problems were identified, both from the scheduling and high level decision-making points of view. The HOLOS architecture was shown in order to give an idea about its philosophy and what is an agile scheduling. We pointed out the importance of developing a DSS as well as on how the agile scheduling system can be integrated in an EE framework. An integration infrastructure architecture was described, which allows a flexible and gradual enterprise integration in the network and, at the same time, guarantees the enterprise independence and privacy. Besides this infrastructure characteristic, we see as other balanced aspects of our general approach the accommodation of legacy systems (both at shop floor and high level systems) in the global architecture, and the exploitation of the human experience and flexibility in the DSS.

New international projects in which our Group is now involved will offer an adequate framework to apply our ideas in real cases. On the other hand, considering that the EE concept is quite new, many problems are still open / need to be better investigated.

ACKNOWLEDGEMENTS

We thank New University of Lisbon and UNINOVA for the general infrastructure, CNPq - The Brazilian Council for Research, and Alexandra Pereira Klen for the discussions about distributed business processes aspects.

6 REFERENCES

AMICE (1993) CIMOSA : Open Systems Architecture for CIM, Springer-Verlag, Berlin.

Browne, J.; Sackett, P.; Wortmann, J. (1995) Future manufacturing systems : Towards the extended enterprise, Computer in Industry, Special Issue on Cim in the Extended Enterprise, **25-3**, 235-54.

Cabletron Systems (1995) USA.

Davis, Randall; Smith, Reid (1983) Negotiation as a Metaphor for Distributed Problem Solving, Artificial Intelligence, **20**, 63-109.

Fowler, J. (1992) Proposal for the STEP Data Access Interface Specification, STEP Implementation Specifications Committee, NIST, USA.

Ge, Qing; Findler, N. (1988) Distributed Planning and Control for Manufacturing Operations, Lecture Notes in Artificial Intelligence, (eds. C.J.Barter & M.Brooks), 440-7, SpringerVerlag.

Hitchcock, M (1994) Virtual Manufacturing - A Methodology for Manufacturing in a Computer, Proceedings of Wokshop on The Automated Factory of the Future : Where do we go from here ? / IEEE 1994 Int. Conference on Robotics and Automation, San Diago, USA.

Huhns, M., editor (1987) Distributed Artificial Inteligence, Pitman Publishing / Morgan Kaufmann Publishers, San Mateo, USA.

KQML : Knowledge Query and Manipulation Language, ARPA Knowledge Sharing Effort Project.

Mackiewicz, R. (1994) An Overview to the Manufacturing Message Specification, in http://litwww. epfl.ch/~mms.

NIIIP Reference Architecture (1995) Concepts and Guidelines, Report N° NTR9501, USA.

O'Neill, H. (1995) Decision Support in the Extended Enterprise, Ph.D. Thesis, Cranfield University.

Osorio , A. L.; Camarinha-Matos, LM. (1993) Information based control architecture for CIM, IFIP Conference Towards World Class Manufacturing, Phoenix, USA.

PRODNET (1995) Production Planning and Management in an Extended Enterprise : Research Feasibility Study Report, ESPRIT Project 20568.

Rabelo, R.; Camarinha-Matos, L.M. (1995) A Holistic Control Architecture Infrastructure for Dynamic Scheduling, Artificial Intelligence in Reactive Scheduling, (ed. Roger Kerr and Elizabeth Szelke), Chapman & Hall, 78-94.

Rabelo, R.; Camarinha-Matos, L.M. (1996) Deriving Particular Agile Scheduling Systems using the HOLOS Methodology, to appear in Journal of Informatics and Control, **5-2**.

Sandholm. T.; Lesser, V. (1995) On Automated Contracting in Multi-enterprise Manufacturing, Computer Science Department, University of Massachusetts at Amhest, USA.

7 BIOGRAPHY

Mr. Ricardo J. Rabelo received his degree on Computer Science in 1984, worked as consultant for several Brazilian companies as a collaborator of GRUCON / Federal University of Santa Catarina, and he is actually finishing his Ph.D. at New University of Lisbon / UNINOVA on Robotics and CIM. His main interest are : dynamic scheduling and virtual manufacturing.

Dr. Luis M. Camarinha-Matos received his Computer Engineering degree and Ph.D. on Computer Science (Robotics and CIM), from the New University of Lisbon. Currently he is auxiliar professor (eq. associate professor) at the Electrical Engineering Department of the New University of Lisbon and leads the group of Robotics Systems and CIM of the UNINOVA's Center for Intelligent Robotics. His main research areas are : CIM systems integration, Intelligent Manufacturing Systems, and Machine Learning in Robotics.