# **Support for Concurrent Engineering** in CIM-FACE

A. Luis Osório; L.M. Camarinha-Matos Universidade Nova de Lisboa Quinta da Torre - 2825 Monte Caparica - Portugal Tel:+351-1-2953213 Fax:+351-1-2957786 E-mail {lo,cam}@uninova.pt

#### Abstract

A federated architecture for Concurrent Engineering is presented and special emphasis is put on the aspects of enterprise modelling and control knowledge to support the coordination of teams of experts. The prototype described combines both information integration and cooperation support functionalities. This paper addresses mainly the business plan interpreter and its multilevel supervision knowledge. Open questions and directions for further research in the context of networks of enterprises (extended or virtual enterprise) are summarized.

#### Keywords

Systems Integration, Concurrent Engineering, CIM, Federated Architecture, Modeling.

#### 1 INTRODUCTION

The concept of lean / agile manufacturing is a result of the increasing globalization of the economy and openness of markets and the tough challenges this situation imposes to manufacturing companies. One of its manifestations is the recognition of the product, and thus product data, in its entire life cycle, as the main "focus of attention" in the CIM Information System. Product Data Management may be considered an essential set of tools for tracking products from conception / design to retirement / recycling. The concept of Concurrent Engineering (CE) is thus the result of the recognition of the need to integrate diversified expertise and to improve the flow of information among all "areas" involved in the product life cycle. Team work based on concurrent or simultaneous activities, potentially leads to a substantial reduction in the design-production cycle time, if compared to the traditional sequential "throw it over the wall" approach.

Evolving from earlier attempts, represented by the paradigms of "Design for Assembly / Design for Manufacturing", Concurrent Engineering is thus a consequence of the recognition that a product must be the result of many factors, including:

-Marketing and sales factors

-Design factors

-Production factors

-Usage factors (intended functionalities / requirements)

-Destruction / recycling factors.

For all these areas there are hundreds of computer-aided tools (CAxx) on the market that help the human experts in their tasks. At a particular enterprise level various of these tools may be available, together with some proprietary software developments. A platform that supports the integration of such tools (information and knowledge sharing) as well as the interaction among their users (team work) is a computational requirement for CE.

© Springer Science+Business Media Dordrecht 1995

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

Observing companies' evolution in terms of organization, a strong paradigm shift towards team-based structures is becoming evident. Team work, as a practical approach to integrate contributions from different experts, by opposition to more traditional hierarchical / rigid departmental structures, is being extended to all activities and not only to the engineering areas.

À realistic approach to design an architecture that supports Concurrent Engineering has to take into account results and tendencies emerging from various research sub fields of the advanced manufacturing area. The definition of a platform for Concurrent Engineering involves, in our opinion, three related sub-problems:

i) Definition of common models. This is a basic requirement in order to enable communications between members of the engineering team. The adoption of common modeling formalisms is a first requirement. Formalisms like IDEF0, NIAM, Express/ Express-G, Petri nets are being widely used. The consolidation of STEP (ISO 1991) may help in terms of product modeling, but many other aspects not covered by STEP have to be considered, like process and manufacturing resources modeling. MANDATE seems still far from offering usable results. Business Processes modeling, as proposed by CIMOSA (Esprit 1989), is also contributing to facilitate dialogue.

ii) Engineering Information Management. Definition of integrating infrastructures and information management systems able to cope with the distributed and heterogeneous nature of CIM, has been the subject of many research projects from which various approaches and prototypes have been proposed in last years (Camarinha-Matos 1991), (Camarinha-Matos 1993). Management of versions, a difficult problem in engineering data management, is even more complex when different versions may be produced / explored in parallel / concurrent way. Various centralized and decentralized solutions have been experimented, the concept of federated architectures developed and the issue of interoperability between different data management technologies and standards has been pursued (Camarinha-Matos 1994). The need for a more mature technology for Engineering Information Management, combining features from Object Oriented and Knowledge Based Systems, Concurrent / multi-agent systems, is becoming evident.

iii) Process supervision. To build a platform that supports concurrent engineering it is not enough to guarantee that the various computer-aided tools used by a team are able to communicate and share information. In other words, it is not enough to provide an integrating infrastructure and to normalize information models. Even though these aspects are essential, there is also the problem of coordination. It is necessary to establish a supervision architecture that controls or moderates the way and time schedule under which computer-aided tools (team members) access the infrastructure and modify shared information. In other words, it is necessary to model the various business processes and to implement a process interpreter or supervisor.

The platform for integration and concurrent engineering -- CIM-FACE: Federated Architecture for Concurrent Engineering -- being developed at the New University of Lisbon addresses these three issues.

## 2 THE CIM-FACE ARCHITECTURE

Figure 1 illustrates the main blocks of the CIM-FACE architecture. Application (computeraided) tools are integrated via the integrating infrastructure, which provides access to the common Information System (IS).

One important part of the CIM-FACE prototype is the EIMS (Camarinha 1993), (Osório 1994) (Figure 2) subsystem which provides basic information management functionalities as well as an integrating infrastructure to support the connection of a federation of heterogeneous software tools.



Figure 1 CIM-FACE Architecture.

The implemented EIMS prototype, inspired on the developments of STEP/ Express (Schenck 1994), is based on a hybrid and distributed programming environment supporting the connection of tools implemented in UNIX and MSDOS environments. The integrating infrastructure supports two connection modes: tight and loose connection. For tight connection a library of Information Access Methods (IAM) is linked to each tool, thus hiding the communication details (RPCs, messages format, etc.). As this part of the system was developed in an earlier stage, before the availability of the STEP SDAI specification, the IAM methods don't follow the standard, although they are quite similar, as inspired by EXPRESS concepts.

For loose connection mode, a STEP-port based on neutral files interchange is included. Reactive programming was used as an effective mechanism to implement interoperability between different data management technologies. The "visible" part of EIMS is implemented on a KBS (Knowledge Craft), but links to a RDBMS or CAD DB were established to offer object's persistency. The interoperability mechanisms play an important role in the migration of legacy systems to more advanced Engineering Information Management Systems.



Figure 2 Engineering Information Management System.

Another major component of CIM-FACE is the Supervisor or Business Plan Interpreter (Camarinha-Matos 1995), (Osório 1994). The objective is to provide a framework that allows for a set of autonomous agents -- computer aided tools and engineers -- to cooperate in solving complex tasks.

Special tools - Business Plan Assistants - provide a front end to the human experts allowing them to take part in the execution of the business processes.

Each agent may have its own data models (partial views of the world) and its decision making capabilities. For instance, the time window within which a CAxx tool is active, interacting with the federation, and the kind of interaction is decided by the couple "CAxx tool - human user". A global supervision system -- "federal government" -- can impose some rules regarding the interaction, like refusing it if some pre-conditions are not satisfied, but it cannot consider the agents as obedient "slaves". On the other side, as agents are supposed to cooperate, they are not completely independent from each other. For instance, the actuation of a CAPP agent depends on the existence of a product model generated by a CAD agent.

### 3 THE ENTERPRISE FUNCTIONAL VIEW

**Business plan.** CIM-FACE follows the CIM-OSA approach to model the enterprise in terms of hierarchies of Business Processes and Enterprise Activities (Camarinha-Matos 1995), (Osório 1994). At each level of the hierarchy, extended Procedural Rule Sets define precedence constraints between BP/EAs of that level, as well as their starting (firing) conditions. We call this hierarchical structure business plan.

Building up business plans is a task for an <u>enterprise modeler</u>, Figure 3. This expert may resort to some domain knowledge, specific to the target industrial sector or to his company. Such knowledge includes a taxonomy of classes of Business Processes and a catalog of installed functionalities (Enterprise Activities, i.e., the functional model of the particular enterprise).



Figure 3 Edition of the enterprise's business plan.

Considering that a business plan will be performed by a team of experts (using several tools), potentially in concurrency, there are other additional aspects to be modeled. A summary of the tasks for the enterprise modeler are:

- Definition of shared information models, i.e., models common to, and that support the interactions among, team members. These models might include methods to derive particular views of the common concepts.

- Definition of business plans. This activity includes the creation of instances of the taxonomy of BP and their parametrization, definition of PRS, assignment of tools and other resources to the enterprise activities, etc..

- Characterization of the team of experts assigned to the business plan: identification and rights (namely in terms of allowed operations).

It shall be noted that these models are important for the implementation of business plan interpreter. Figure 4, shows a partial view of some concepts that support the Business Plans.

{{ PlanNode: Identifier : Name : InputConcepts : OutputConcepts : PrevActivities :	{{ BusinessProcess: is-a : PlanNode SubObjectives :
NextActivities : UserRules :	
StartActivity : ; methods EndActivity : ShowStatus : ShowSubPlan : SendMsg : ReadMsg : }	{{ Enterprise-Activity: is-a PlanNode : ToolBox :  }}

Figure 4 Basic concepts in a business plan.

**Contexts.** As there might be many business plans running in parallel (supported by different teams), it is important to provide a different context to each business plan. On the other hand, even inside one team / one business plan, some members (or subgroups) might need some "privacy" while they are exploring some rough ideas and before they decide to make their results public to the team. Therefore, CIM-FACE offers the possibility to create hierarchies of contexts. All IS objects created / modified in a given context won't be visible outside this context or its children contexts. The implementation of this concept in Knowledge Craft is quite straightforward.

( Puoinoso Dien:	/ InformationContext
{ DusinessFian.	[ Information Context
HootNode:	BusinessPlan
InformationContext:	ObjectList
UserRestritions:	SubContexts
; methods	; methods
CreatePlanContext	CreateSubContext
EnterContext	ExportContextObjects
ListPlanContexts	DeleteSubContext
}	}

Figure 5 Information Context concept.

When a business plan is started, a root context is associated to it. New child contexts may be created outside the current context provided the team member has enough rights (method CreateSubContext). The new context is created as a subcontext of the current one and normal inheritance rules apply, i.e., all objects visible in a parent context are available to the child context.

In this way, a team member (or subgroup) can explore his ideas without interfering with other members involved in the business plan. Once he is satisfied with his results, he can make them available to the others, i.e., <u>export</u> them to parent context (method ExportContextObjects).

Therefore, we can have several contexts associated (in a hierarchical way) to a business plan. The methods EnterContext and ListPlanContexts are useful to navigate through these contexts.

{{BP012: Type : BP Instance : BP\_design Name : 'Design P1' Input : Output : Product PrevActivities : NextActivities : UserRules : all\_concluded(Self), version(Self, less\_than(5)), ... SubObjectives : EA123, EA124, EA125 Initializer : create instance(Product, Name, NewVersion) }} {{EA124: . Type : EA Instance : Product\_Design\_Analysis Name : 'DesignAnalysis P1' Input : RoughDesign, Output : DesignConstraints PrevActivities : EA123 NextActivities : EA125 UserRules : attribute value(RoughDesign, material, metal), ... ToolBox: PDA1 Initializer : create instance(DesignConstraints,

Name, NewVersion) }}

{{EA123: Tvpe : EA Instance : Product Rough Design Name : 'RoughDesign\_P1' Input : Output : RoughDesign PrevActivities : NextActivities : EA124 UserRules : attribute value(RoughDesign, material, metal), .... ToolBox: CAD1 Initializer : create\_instance(RoughDesign, Name, NewVersion) }} {{EA125: Type : EA Instance : Product Design Name : 'Design\_P1' Input : RoughDesign, DesignConstraints Output : Product PrevActivities : EA124 NextActivities : UserRules : attribute\_value(Product, weight, less than(23)), ... ToolBox: CAD1, FEA1 Initializer : }}

Figure 6 Examples of BP and EA concepts.

**Precedence relationships among activities.** The Procedural Rule Set mechanism of CIM-OSA doesn't seem quite flexible to represent the temporal interdependencies among activities in a context of Concurrent Engineering. For instance consider the activities of "Product Design" and "Process Planning". Although they can proceed with some degree of concurrency (i.e., process planning can start once a first draft of the product is made), Process Planning cannot finish before Product Design finishes. At least some details of the process plan definitely depend on the final commitment regarding the product model. In Figure 6, some rules are used to declare precedences among exemplified Enterprise Activities. Classical precedence relationships don't capture the full semantics of this temporal overlapping situations.

Therefore CIM-FACE considers rules that combine some aspects of PRS with the Allen's temporal primitives.

For example, consider the illustration of figure 7, where four activities are carried on partially in concurrency: A1- launch New Product process, A2 - Rough Design, A3 - Functional Analysis, A4 - Geometrical Design.



Figure 7 Relation among concurrent activities.

In this example, the following rules apply:

- i) Activity A1 must start before all other activities and should finish during execution of A2
- ii) Activity A2 starts after A1 and finishes after A1
- iii) Activity A3 must be done during A2
- iv) Activity A4 starts after A2 and finishes after A2.

The specification of this kind of constraints is very important to drive the business plan interpreter (see next chapter). Such rules are specified by the enterprise modeler.

**Information views.** Although the enterprise information modeling is out of the scope of this paper, for details on the CIM-FACE approaches to information integration and management in (Camarinha-Matos 1993), (Camarinha-Matos 1991) there are some aspects that are strongly related to the supervision of team - based activities.

One aspect is the concept of object view. The common IS contains the concepts / models that are shared by, or support the interaction among team members. But for some of these concepts, different team members will have different views or perspectives.

Therefore, the enterprise modeler(s) is supposed to model the different views and to associate them to the descriptors of the team members. This implementation may include methods to derive (and integrate) the various views from (into) a common model.

In some cases the views / contributions of each member / tool can be considered just as facets (slots) of a global concept (see Figure 8).



Figure 8 Partial views of an Information Object.

Therefore, optional constraints related to the facets "accessible" to each member / tool may be associated to the EA model (Input/Output slots, Figure 6) and to the user model.

# 4. MODEL INTERPRETER

After the enterprise modelling and, in particular, after the definition of a business plan, a model interpreter is necessary to support the operational phase.

The model interpreter is responsible for the "execution" of the various enterprise's business plans. It is responsible for:

-keeping track of the execution status of each business plan (and each node in the plan)

-supporting consistency maintenance (by verifying pre- and post-conditions for each BP/EA) -helping or directing CA-tool selection for each EA, when more than one tool is available for an EA

-providing a common access to the enterprise information models and supporting a multicontext framework

-allowing communication among team members

-providing a platform for progressive improvement of the control structure (addition of new rules).

An important part of the execution environment is the set of CA-tools that - in cooperation with the various team experts - actually implement the EAs. A "protocol" is necessary to specify the interactions between each couple tool-user and the integrated federation. The "performer" of this protocol can be a layer separated from the applicative part of the tool. For legacy systems it is quite hard or nearly impossible to modify their control architectures. For new tools this "protocol performer" can be seen as a common script (library) that can be linked to the tool.

As a first attempt, a tandem structure (Figure 9) was implemented, separating the tool itself from the protocol performer, here called *Business Process assistant*. From the implementation point of view, this tool assistant can be a module linked to the tool or a parallel (detached) process. The second alternative is more suited to legacy systems.

With basis on this initial approach, but taking into account that:

-the "tasks" to be realized in a given business plan are hierarchically decomposed

-"real" application tools appear only associated to the leaves of this hierarchy (EAs)

-the human experts may interact with (inspect, start, finish, etc.) activities at various levels of the hierarchy.

The initial concept of tool assistant (Osório 1994) evolved to the concept of business plan assistants (Camarinha-Matos 1995).



#### Figure 9 Business Plan Assistant.

Whenever a user "joins" the federation (logs in), a business plan assistant is launched. Through a graphical interface (Figure 10), the user has access to a specific business plan (or part of it, according to his access rights). For each node in the plan he can inspect its status, start or stop the BP/EA, create or change the default context, etc.. When a valid team member wants to enter into the CIM-FACE environment (Login) he must enter a login identification, like in classical multi-user computational platforms. This identification is used by the CIM-FACE system to condition the access to different levels of interaction and supervision rights.

As a result of login, the user is positioned in the execution context of one or more activities of the selected business plan. From that plan, the human can select, for instance, "StartActivity" for an activity not being executed (if he has enough rights). Depending on the confidence level assigned to this user, by the enterprise modeler, the interpreter can interdict the execution if some validation rule fails. When starting an EA, the associated application tools are launched.

Therefore, the human expert plays an important role in the evolution of the business plan execution.

As the control events are decided externally, by the human, how can the process interpreter be sure the decision was appropriate? For instance, lets suppose the current user of tool  $T_i$  informed the system that he finished the generation of a process plan for a given production task. Should the control system simply accept such information as a fact or should it be cautious and try to investigate the accuracy of the information? Therefore different "kinds" of control systems can be defined, ranging from a totally confident system to a cautious one.

The effectiveness of this control depends on the "quality" of the models defined in previous chapter.





**Characterization of supervision knowledge.** From the concurrent engineering point of view, the execution of an activity should be flexible enough to facilitate engineers the exploration of different alternatives or even the redefinition of a previous approach regarding a new idea received from other cooperating team member.

It is necessary however to provide the system (task for the enterprise modeler) with mechanisms to express constraints to the execution of planned activities. Taking into account the fact that the execution environment is strongly based on a distributed infrastructure, maybe even located in different enterprises, various levels of validation can be considered. Not all of these levels are direct responsibility of the plan interpreter. We think that control knowledge may be defined at three levels:

1. Information System level - constraints may be embedded into the common information models. This can be represented by local and global rules of EXPRESS. The execution of such rules is the responsibility of the Information Management System.

2. Application level. Various control aspects, due to their specificity, are better defined inside specific application software (enterprise's proprietary tools).

3. Business plan level. Rules like precedence constraints and other specific pre- and postconditions may be associated to each BP/EA node. The business plan interpreter has mainly to take this control into account.

In some cases, and for some application domains, it will be possible to define a set of verification rules to test the validity / accuracy of each access protocol action issued by business plan assistants. In other cases that might be difficult. Therefore, our proposal is to have an architecture that can start from a level of total confidence and progress towards a more cautious system once verification rules are added to its control knowledge base.



Figure 11 Check rules associated to BP/EA.

```
An example for a minimal checking level is a rule that implements a CIM-OSA PRS rule:

IF <all previous activities are finished with success>

and <Self was requested to start>

THEN <enable execution of Self>

A more specific example can be associated to the BP_Design_P1 of type BP_design:

IF <MarketEvaluation_p1 of type MarketEvaluation was started> and

<BP_Design_P1 was requested to start>

THEN <enable execution of BP_Design_P1>

For instance the following rule could be attached to the EA124 (Figure 6):

IF <RoughDesign.verified of EA123 Output is true> and

<EA123 is inactive> and <Self was requested to start>

THEN <enable its execution>.
```

It shall be noted, however, that rules associated to the nodes (BP/EA) of a business plan are related to the coordination / interaction level of the plan execution. Other kind of rule, related to information consistency (first level mentioned above), are more appropriately defined in association to the concepts present in the IS. See, for instance, the following example using Express notation. This class of rules are supposed to be checked by the Information Management System and not by the Business Plan Interpreter.

```
ENTITY manufacturing_step;
  identification : IDENTIFIER;
  manpower_time : INTEGER;
  machine_time : INTEGER;
  cycle_time : INTEGER;
   . .
WHERE
  rule_1: cycle_time >= machine_time;
  rule_2: cycle_time >= manpower_time;
END ENTITY:
```

Messages inter team members. As various BP/EA nodes may be active (in parallel), i.e., "operated" by different humans, possibly in different geographical locations, and a change in one may affect the progress of others, it is important to provide a message exchange mechanism. This can be seen as a particular e-mail facility associated to the plan nodes. From

one BP/EA node, the human agent can send a message to another BP/EA (or to a group). See (Figure 10). At current stage, these messages are intended for communication among the human experts, but we may even think of messages to be processes by high level (autonomous) process execution. In this way, if an agent modifies the model of a part, besides making the new model available in a common context, he can notify the agent performing the process plan, which may need to revise his plan.

#### 5 CONCLUSIONS AND FUTURE DEVELOPMENTS

CIM-FACE represents a prototype federated architecture to support systems integration and concurrent engineering activities.

The implemented parts include:

i) an infrastructure for information integration and management that was successfully evaluated in the context of an European Esprit (CIM-PLATO) project.

ii) A process interpreter that provides basic functionalities for concurrent execution and coordination of a set of business processes / enterprise activities, driven by a team of human experts. The proposed architecture supports the definition of multiple checking levels, allowing progressive degrees of robustness.

An intensive evaluation of this control strategy in a real application has still to be done. CIM-FACE is not a finished system, but an ongoing research.

A future extension of this work has to do with the concept of extended or virtual enterprise. There is a tendency to establish partnership links between companies, namely between big companies and networks of components' suppliers. Similar agreements are being established between companies and universities. Such network structures may be seen as extended or virtual enterprises. In fact, the manufacturing process is not anymore carried on by a single enterprise, but each enterprise is just a node that adds some value (a step in the manufacturing chain).

This tendency creates new scenarios and technologic challenges, specially to Small and Medium Enterprises (SMEs). Under classical scenarios, these SMEs would have big difficulties -- due to their limited human and material resources -- to have access to or to use state of the art technology. Such partnerships facilitate the access to new technologies and new work methodologies but, at the same time, impose the use of standards and new quality requirements.

The efforts being put on the implantation of high speed networks (digital high ways), supporting multimedia information, open new opportunities for team work in multi-enterprise / multi-site networks. But this new scenario also brings new requirements in terms of control: access rights to the information, scheduling of access, control of interactions, etc. (distributed information logistics infrastructure).

While in recent past the emphasis was put on enterprise integration, now the challenge is to provide tools that support an effective management, engineering and control of production in a framework where each node has a large autonomy, operates under different philosophy, and sometimes has competing objectives with respect to other nodes.

A main aspect is the flows between nodes and the monitoring of their evolution. Several aspects may be considered:

i. Inter enterprise engineering developments

iii Efficiency of orders flow

iii. Follow up of orders evolution

iv. Distributed and dynamic scheduling

v. Incomplete and imprecise orders

vi. Network-wide workload optimization

The extension of the federated approach of CIM-FACE to this scenario of network of enterprises is our next plan.

### 6 ACKNOWLEDGMENTS

The work here described received partial support from the European Community -- the Esprit CIM-PLATO, ECLA CIMIS.net and FlexSys projects -- and from the Portuguese Agency for Scientific and Technologic Research (JNICT) -- the CIM-CASE project.

# 7 REFERENCES

- Camarinha-Matos, L.; Sastron, F. (1991)- Information Integration for CIM planning tools, CAPE'91 - 4th IFIP Conference on Computer Applications in Production and Engineering, Bordeaux, 10-12 Sep, 1991.
- Camarinha-Matos, L.M.; Osório, A. L. (1993) CIM Information Management System: An Express-based integration platform, IFAC Workshop on CIM in Processes and Manufacturing Industries, Espoo, Finland published by Pergamon Press.
- Camarinha-Matos, L.M.; Osorio, A.L. (1994) An integrated Platform for Concurrent Engineering, Proc.s 3rd CIMIS.net Workshop on Distributed Information Systems for CIM, Florianopolis, Brazil. To appear in the Journal of the Brazilian Society of Mechanical Science.
- Camarinha-Matos, L.M.; Afsarmanesh, H. (1994) Federated Information Systems in Manufacturing, Proceedings of EURISCON'94, Malaga, Spain.
- Camarinha-Matos, L.M.; Osório, L. (1995) CIM-FACE: A Federated Architecture for Concurrent Engineering, CAPE'95, IFIP International Conference on Computer Applications in Production and Engineering, Beijing, China, May 1995.
- Esprit Consortium AMICE (1989) Open System Architecture for CIM, Springer-Verlag.
- Osório A. Luis; Camarinha-Matos, LM. (1994). Information based control architecture, Proceedings of the IFIP Intern. Conference Towards World Class Manufacturing, Phoenix, USA, Sep 93, edited by Elsevier - North Holland.
- Schenck, Douglas; Wilson, Peter (1994), Information Modeling the EXPRESS Way, Oxforf University Press, New York.
- STEP, ISO (1991). Reference Manual, ISO/TC 184 /SC4.
- Welz, B. G. et al. (1993). A toolbox of integrated planning tools a case study, IFIP Workshop on Interfaces in Industrial Systems for Production and Engineering, Darmstadt, Germany, 15-17 Mar 1993.

# 8 BIOGRAPHY

Eng. A. Luis Osório received his BSc on Electronic and Telecommunication from Polytechnic Institute of Lisbon, ISEL and Computer Engineering degree on Computer Science, from the New University of Lisbon. Currently he is adjunct professor at the Electronic and Communications Engineering Department of the ISEL. He is also researcher and PhD student in the group of Robotic Systems and CIM of the UNINOVA's Center for Intelligent Robotic. His main research interests are: Information based integration and control of CIM systems, Intelligent Manufacturing Systems, Artificial Intelligence.

Dr. Luis M. Camarinha-Matos received his Computer Engineering degree and PhD on Computer Science, topic 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 Robotic Systems and CIM of the Uninova's Center for Intelligent Robotics. His main research interests are: CIM systems integration, Intelligent Manufacturing Systems, Machine Learning in Robotics.