IMS Project 99009

PABADIS
Plant automation based on distributed systems

Final IMS Project Report

This final project report for the IMS project 99009 (PABADIS) is, on behalf of the PABADIS consortia, respectfully submitted to IMS by the international co-ordinating partner.

As explained in the document, this project was carried out in close collaboration between four IMS regions, Canada, Europe, Switzerland and United States.

IMS: (Intelligent Manufacturing Systems program)

Rev.	Content	Resp. Partner
1.0	Document contents compilation and editing	The PABADIS Consortium Phoenix; Eckehardt Klemm

Final approval	Name	Partner
Reviewer	Dr. Axel Klostermeyer Dr. Arndt Lüder	IAF IAF

Executive Summary

This report provides an overview over the inter-regional IMS project achievements. This report shall provide a comprehensive view of the results obtained, the methodologies and approaches employed, changes in the state-of-the-art since the project was started. This report addresses the objectives for the project as well as the degree to which these objectives have been reached.

The main concept, research and development work has been carried out by the EU project, the US and the Swiss project provide substantial contributions to the results of PABADIS. Unfortunately the Canadian project eventually failed to get funded and could only contribute little to the project. In spite of this problem the high personal commitment of the active researchers and regional coordinating partners was of great help for the progress of the project and the work could be completed with a successful prove of concept phase.

By and large, the international collaboration was needed and contributed positively to the results of PABADIS. Equally important is, that the PABADIS concept and achievements has been brought to a wider R&D community and in that way the results will outlast the end of the primary PABADIS project and – hopefully – fertilise future plant automation concepts.

Summarising, this report contains a short introduction to the project itself, the partners involved, and the main achievements in chapter 1.

Chapter 2 contains the project objectives, embracing as well technical as management issues. Though the project needed a prolongation of about 8 months, it can be stated that the original project objectives were reached and at some points even beaten. This was mainly due to a rather balanced consortium.

Chapter 3 then reports about the methodologies applied and the lessons learnt. For example it became clear that a classical waterfall model does not fit the requirements of a modern IT project; thus, it was enhanced by means of a spiral model. Modern IT technologies as a BSCW server were made use of.

Chapter 4 then gives a detailed description on project results and achievements. As the PABADIS project focused on the development of a general concept for a high flexible production architecture using new technologies such as agents and Plug-and-Participate-technologies, the scientific and technological outcome of the project was on the one side a detailed specification of such a system and on the other side a proof-of-concept for the applicability of the mentioned technologies. This includes the implementation of software libraries for all basic components of a PABADIS-system and the development of different prototypes to show the functionality for the intended production scenarios.

In chapter 5 the deliverables and references produced during the project's run time are shortly introduced.

Chapter 6 gives an overview about project management and co-ordination aspects. This embraces in detail a short overview about project management as such, a statement about performance and dedication of the consortium, the challenges encountered, co-operation with industry, other projects and dissemination aspects, and finishes with dissemination activities. It can be stated that no extraordinary challenges arose and the performance of the consortium overall was very well.

Chapter 7 provides an outlook on the future work of the partners with regard to PABADIS. Here especially the exploitation plans of the single consortium members are listed.

Finally, the project's management team would like to thank IMS for its international project framework, the European Union and the Swiss government for funding this project.

Last but not least, the management would like to thank, of course, the PABADIS consortium itself. The achieved results – which are more than originally intended – would not have been possible without a very strong dedication to the Project's contents and a full support to the idea of IT based, distributed automation.

Contents

1 Project Overview **6**

1.1 Introduction to the Project itself 6

1.2 Consortium Composition 6

1.3 Contribution of the different regions to the project 7

1.4 Main Achievements of the Project 8

2 Project Objectives **9**

3 Methodologies **11**

4 Project Results and Achievements **12**

4.1 Definition and Design 12

4.2 Development 19

4.3 Testing 29

5 Deliverables and References **33**

6 Project Management and Co-ordination Aspects **40**

6.1 Overview 40

6.2 Performance and Dedication of the Consortium 40

6.3 Encountered Challenges 41

6.4 Co-operation with Industry, other Projects, and Dissemination 41

6.4.1 The PABADIS Industrial Reference Group 41

6.4.2 Consideration of related IMS projects 42

6.4.3 Consideration of related EU Projects 42

6.5 Technology transfer and dissemination 43

7 Outlook **43**

8 Conclusions **44**

9 Annex **45**

9.1 Dissemination 45

9.1.1 Journal papers and book contributions 45

9.1.2 Conference papers 45

9.1.3 Presentations without conference proceedings, exhibitions 48

9.1.4 Conference Work Shop Organisation 48

9.2 IMS Project Meetings 49

List of Figures and Tables

Figures

Figure 1: Steps of a life cycle of a product processing in the PABADIS-system. 15
Figure 2: General structure of a PABADIS-agent 16
Figure 3: PABADIS agent class hierarchy 17
Figure 4: Communication relations between the lookup service and the various software agents 19
Figure 5: General implementation structure of a CMU 21
Figure 6: Distribution of CMU parts 21
Figure 7: The functionality of the RAFI within the PABADIS system 23
Figure 8: Generic ERP-PABADIS Interface 24
Figure 9: Structure of the ERP – Agency interface 25
Figure 10: Agency architecture 26
Figure 11: Structure of the Fischertechnik-model demonstrator 30
Figure 12: Structure of the experimental plant 30
Figure 13: IMS project management structure 40

Tables

Table 1: Project partner overview 7
Table 2: Distribution of MES-functions 14
Table 3: Related Projects 42
Table 4: Related Projects 42

1 Project Overview

1.1 Introduction to the Project itself

Modern markets are both global and turbulent. In this environment, retaining a customer focus means becoming competitive across an ever widening range of specific products and services. Economists have recognised this trend and have developed several organisation and management concepts which are generally summarised by the term “Mass Customisation”. In order to implement the various elements of “Mass Customisation”, firms must also adapt their technical resources correspondingly. Hence, technology and system architecture adaptation must follow organisational change. Though there is a far-reaching agreement concerning this point, consequent practical architectures, solutions, or implementations are still lacking.

Against this background, this chapter will present an overview of the IMS research project PABADIS. PABADIS means Plant Automation Based on Distributed Systems and focused on automation in one piece production plants using distributed control systems. This new approach’s objective was to create a plug-and-participate environment which allows production companies (a) to simply plug in a new machine and use it without major changes within the legacy systems and (b) to make job control more flexible by augmenting “conventional” (main-stream) Enterprise Resource Planning (ERP) functionality with intelligence inherent in mobile agents. This lead to the realisation of the following vision:

“A future plant will consist of machines and systems which have their own de-centralised intelligence. Using plug-and-participate-technology, these machines will be able to connect by their own means to a control network. There will be the least amount of centralised control as possible. The only supervision to these machines will happen within an agent supporting interface to the conventional ERP system. Maximum use will be made of (software) agents which are highly flexible and can be reconfigured on the fly, in response to new situations or demands. This will lead to a highly flexible and adaptable plant as an appropriate technical response to a firm’s adoption of today’s modern management and organisation concepts”.

The consortium which designed, developed, and implemented the corresponding architecture is described in the following sections:

1.2 Consortium Composition

From the start of PABADIS there have been activities to set up the IMS project. In the beginning there have been partners from EU, Japan and US. The first attempt was to build an European-, an Japanese- and an US-Project, but there have also been activities to form a Swiss project. Unfortunately we did not find Japanese industrial partners under the economic situation at that time and due to the procedures in Japan, which only recognise industrial partners as regional co-ordinator, it was not possible to set up a Japanese IMS project.

Therefore we had to reconsider the project structure and finally decided that the project should be accomplished with four regional projects - Canada, EU, Switzerland and US. However it took much longer than expected to finalize the consortium agreement and get it signed by all partners. This problem delayed the start of the PABADIS IMS project significantly and final proposal was delivered to IMS in June 2002 and the proposal was endorsed in August 2002.

The final consortium (see following table) is a good balance between academic and industrial partners from the different regions. However, the European project part holds the majority.

Company	Address	Type	Role
A. Hatzopoulos	Hatzopoulos S.A. - Thessaloniki, Greece	I	European project partner
aJile Systems	aJile Systems, Inc. - San Jose, CA, USA	I	US project partner
ALTEC	ALTEC S.A. - Thessaloniki, Greece	I	European project partner
Unige-CUI	Centre Universitaire d'Informatique Université de Genève Genève, Switzerland	A	**Swiss project coordinating partner**
EMA	EMA Site EERIE Service Communication Nimes, France	A	European project partner
IAF	Otto-von-Guericke University, IAF-FMB Magdeburg, Germany	A	**European project coordinating partner**
ICT	Technische Universität Wien Institut für Computertechnik Wien, Austria	A	European project partner
IMS GmbH	IMS GmbH - Magdeburg, Germany	I	European project partner
Jetter AG	Jetter AG - Ludwigsburg, Germany	I	European project partner
P2I	P2I Engineering - Nimes, France	I	European project partner
PebbleAge	PebbleAge S.A. - Geneva, Switzerland	I	Swiss project partner
Phoenix	Phoenix Electronics GmbH Bad Pyrmont, Germany	I	**Interregional project co-ordinating partner (ICP)** European project partner
PsiNaptic	PsiNaptic Inc. Calgary, Alberta, Canada	I	Canadian project partner
PUM-I	Philipps-Universität Marburg Institut für angewandte Informatik Marburg, Germany	A	European project partner
SIG Pack	SIG Pack Systems AG Beringen, Switzerland	I	Swiss project partner
Sun	SUN Microsystems Burlington, MA, USA	I	**US project coordinating partner**
University of Calgary	Department of Electrical and Computer Engineering; Faculty of Engineering The University of Calgary Calgary, Alberta, Canada	A	**Canadian project co-ordinating partner**
ZHW	Zürcher Hochschule Winterthur Institut für Mechatronische Systeme Winterthur, Switzerland	A	Swiss project partner
Legend: I := industrial partner A := academic partner			

Table 1: Project partner overview

1.3 Contribution of the different regions to the project

The EU project started first and therefore the EU partners developed most of the basic PABADIS concept and covered the main research and development topics of the project. The project results of the European project are documented in detail within the EU project deliverables. A summary of the aims, objectives, and reached research results of the PABADIS project is given in document “Revolutionising Plant Automation - The PABADIS Approach (EU project deliverable Task 6.3)”. In that document the used technologies, the PABADIS system components, the PABADIS system behaviour and the system basic function are explained in detail. This work was the backbone of the IMS project and the other regions contributed on it. In the future this document will be the basement for a book publication on the PABADIS approach and thereby world wide available.

The US project, esp. SUN contributed considerable in the early phases of the PABADIS project as a technology provider for the platform independent programming technology Java and the plug and participate technology Jini. The PABADIS Java concept for limited devices and the plug and participate technology applied were jointly designed between US and European project partners. AJile contributed as provider for Java based field control technology. The PABADIS Java based control devices have been designed jointly between US and European project partners. In both cases the US partners have contributed valuable information to the final project results.

The Swiss partners of the PABADIS consortium assessed and examined the industrial relevance of the PABADIS concept for different manufacturing organisations, focusing mainly on the fast moving consumer goods industry. The relevance assessment covers market research, general observations on PABADIS and an evaluation of the applicability of Java on shop floor. The group concluded, that PABADIS provides a valuable input to the problem domain of flexible manufacturing, but does not fully meet the requirement of the evaluated industry group and is therefore not yet directly applicable. The conclusion is mainly based on the contrary tendency of development and the fact that Java is currently not an accepted technology. In cooperation between European and Swiss project partners the Java based control technology has been advanced. Thereby the involved partners tend to enlarge the acceptance of Java in automation to bring the PABADIS results to application. In addition to the evaluation, within the Swiss project an agent based behaviour modelling was developed in order to provide the PABADIS multi agent system with explicit behaviour.

The Canadian partners contributed with their knowledge of Java based agent platforms, Plug-and-Participate technologies, and Communication systems to the other IMS partners. However, the Canadian project failed to get the expected funding and eventually ceased the research work. But the Canadian project partners were able to establish a cooperation between the PABADIS project and the FIPA organisation, the most recent standardisation organisation for agent technologies.

Note: The failure of the Canadian project due to available funding was unexpected because there are funds for IMS projects in Canada. Unfortunately there was no time to apply again for funding to ensure an reasonable contribution of the Canadian project and we decided to leave the contribution on the level of knowledge exchange. In August 2003 there has been a meeting with the Canadian IMS secretary Alan Martell to discuss the reasons why the Canadian project has failed. We concluded that this bitter experience was to some extend due internal Canadian problems but there are also IMS based reasons. Alan Martel will use our findings within the IMS organisation to improve the IMS project framework.

1.4 Main Achievements of the Project

The main technical achievements of the project are listed below in form of an enumeration:

- A new control architecture based on distributed intelligence, applying agent and plug-and-participate technology, providing:
  - * improved flexibility,
  - * improved robustness,
  - * reduced system design efforts.
- New generic interfaces between MES level and machine level with a standardised way of accessing generic machine functionalities;
- A new product and production process data description technology;
- New generic interfaces between ERP level and MES level;
- New Java based field control technology; and
- A new technology for agent based flexible workflow management.

Corresponding details and explanations will be found within the next chapters of this report.

2 Project Objectives

The PABADIS project focused on automation in one piece production plants using distributed systems. Making use of software agent technology and plug-and-play mechanisms the project aimed at creating a plug-and-participate environment which allows producing companies (a) to simply plug in a new machine and use it without major changes within the legacy systems and (b) to make job control more flexible by augmenting "conventional" (mainstream) ERP functionality with intelligence inherent in mobile agents. This can be summarised by the vision from chapter 1.1:

Generally speaking the scope of the project was the information technology which controls the plant as a whole. This technology is usually found within ERP, MES, SCADA and PLC, NC, RNC, etc.

Currently this is done by a hierarchical structure which consists of the ERP system at highest level, the MES and SCADA systems in between and the PLCs, NCs, etc. at the lowest level. PABADIS aimed at replacing the MES part within this architecture by the innovative DFT (Device Function Technology) concept which has been proven as very efficient and effective.

Based on this concept PABADIS' scientific and technological objectives were:

1. Definition and Design

1.1 Determine the general requirements to use distributed control within an automated production plant.

1.2 Define the overall architecture for plant automation with plug-and-participate features of emerging connection technologies such as Jini™.

1.3 Convergence towards an architecture that enables the direct job control via an ERP system without MES and SCADA systems.

1.4 Design an adequate platform for mobile and residential software agents, upon which multiple-vendor solutions may be provided, which may address highly differentiated manufacturing processes and practices existing in the various end user environments.

1.5 Definition of possible strategies and functions of a single agent which executes a special job.

2. Development

2.1 Develop the plug-and-participate concept and test its technical feasibility for machines and systems. Develop a bottom-up conceptualisation for functions of industrial machines that will be enabled to interact with each other and the company's ERP system using plug-and-play concepts.

2.2 Design generic platforms to enable the communication with mobile agents which can be used within any future PLC, CNC and RNC (vendor independent).

2.3 Develop generic platforms which can be used with any future ERP system to enable the flexible creation of job controlling mobile agents (vendor independent).

2.4 Program generic software platforms for mobile agents in production plants. In this way, standardisation of metadata for component (object) design and reuse may be facilitated, for independent industrial IT vendors; according to this pattern, a component model specifies agent interfaces and describes the design, assembly, and deployment of those components into a system, based on standard component architectural style to be devised within the project lifetime. The new model shall provide the necessary metadata types for these descriptions. The work builds on a specialisation of a UML-based methodology for end-to-end component-based development in industrial environments.

3. Testing

3.1 Realise a test case for the developed platforms within a real industrial surrounding.

3.2 Verify this test against industry's requirements

4. Exploitation

4.1 To evaluate the test case and extract all useful lessons and experiences,

4.2 To disseminate all knowledge gained to all relevant actors, to the external business and academic world, and to standardisation projects (e.g. STEP, Hümnos, OPEN Control, industrial and service companies from outside the project),

4.3 To prepare the ground for post project commercial exploitation using the mobile agent IT infrastructure developed within the project.

Based on these objectives, PABADIS significantly enhances enterprises' abilities to react to market changes and through providing a plug-and-participate environment enhances the efficiency in processing. This covers the reaction on changing market environments, IT flexibility to handle the new logistic requirements or, on the more operative side, optimises the enterprises` job control. Meeting company needs provides consequent gains of:

- Reducing the set up time of new machines and systems. This includes the time it takes to implement changes within an existing company.
- Reducing costs associated with the set-up by reducing the set-up time.
- Prolongation of machine life cycles by enabling the use of these machines within a changing environment.
- Enabling a flexible degree of automation by easy possibilities to in- or exclude special machinery.
- Simplification of the process control since planning and WIP are reduced through DFT. Planning and scheduling can be replaced by a direct monitoring according to the real customer demand.
- Increasing the flexibility and efficiency concerning job control: By building agents which incorporate different strategies in job control.
- Reducing the throughput time by more efficiency in job control.
- Improvement of on time shipment by more efficiency in job control.
- Increasing the flexibility for all tasks concerning the collection of information and the influencing of process data. By developing a platform for mobile software agents for manifold tasks (e.g. collecting quality data) it will become possible for example to monitor a quality critical product entirely different from a mass product.

The above stated advantages can be exploited by the end-using companies as well as by the machine and control manufacturers and plant erectors. This project enables European manufacturers to stay competitive within this market and to gain advantages by the offered advantages for the end-users.

Furthermore, and taking into account existing and emerging trends with regard to the integration of knowledge management and data warehousing in industrial enterprises, the integration of user collaboration, data/info management and machine intelligence requires the integration of many different distributed data sources and software services. PABADIS allows the transition to a highly skilled labour force, consisting of sophisticated workers who access a variety of plant information services more easily and supports the use of standard manufacturing practices and working patterns.

3 Methodologies

Within the PABADIS Project the developments made are based on, on the one hand, standard development strategies exploited from information sciences and, on the other hand, on state-of-the-art technologies and approaches for data management, modelling, communication and interaction.

The applied development strategy was the waterfall model, which was mainly induced by the working plan. Since the application of the strategy is known as very dangerous and error-prone, it was enhanced in the work plan by a spiral model based on the extension of prototypes by additional functions and technologies. Thereby the project consortium first has developed requirements to the project results, has made a system specification and design within a second step, an implementation in a third, and a testing in the final step. Step 3 and 4 and partially step 2 have been made in a cyclic fashion following the spiral model. During the project the PABADIS consortium has seen that the feedbacks from reached implementation and testing results to design decisions was not strong enough. Especially the possibilities of integration of necessary improvements to the overall system design were limited. This was one of the main reasons, which have lead to the project prolongation.

As conclusion for further projects the consortium has determined the necessity of a stronger interlocking of design, implementation, and testing phases to reach a complete spiral model within the project.

Within the project the Project consortium has used a shared common workspace as state-of-the-art technology for data management and data exchange. Here a BSCW system from Fraunhofer GMD has been applied. The consortium has reached very good experiences with this type of technology. It was very useful and has improved the collaboration among project partners especially during the implementation phase.

The communication among project partners beyond direct meetings and the use of the BSCW system has been performed mainly using Internet technologies. Here the ordinary mail and a private Internet chat provided by PUM-I have been used. Particularly the Internet chat has been regularly used for the discussion of design decisions and necessary activities among the project partners. For further projects the application of such an Internet chat can be strongly suggested to improve the project’s performance.

During the development phase the project consortium has used UML as main modelling technology. The consortium has agreed on using the UML tool “Poseidon for UML” from Gentleware. Based on this the consortium was able to design the system on a common modelling framework within a shared modelling process. This was very useful especially for the integration of the different knowledge bases of the different project partners.

4 Project Results and Achievements

As the PABADIS project focused on the development of a general concept for a highly flexible production architecture using new technologies such as agents and PnP-technologies, the scientific and technological outcome of the project was on the one side a detailed specification of such a system and on the other side a proof-of-concept for the applicability of the technologies. This includes the implementation of software libraries for all basic components of a PABADIS-system and the development of different prototypes to show the functionality for the intended production scenarios.

In the following the main results of the project will be discussed in relation to the objectives mentioned in section 2.

4.1 Definition and Design

Objective 1.1: Determine the general requirements to use distributed control within an automated production plant.

Within the project a general concept for a distributed control system for an automated plant was developed which based on an detailed analyses of existing industrial requirements. The software architecture of this system follows the approach to distribute all relevant functionality for job control among the plant and uses the agent paradigm as a basic concept for this. Therefore basic rules for the distribution of functional units in the system were defined. This leads to an architecture where classical MES-level between ERP and field level is replaced by a community of agents responsible for the workflow management. The resources of the production system are represented by so-called Co-operative Manufacturing Units, while mobile agents are responsible for the execution of production orders and the management of the production system.

Objective 1.2: Define the overall architecture for plant automation with plug-and-participate features of emerging connection technologies such as Jini™.

Based on an agent-oriented design, a PABADIS system basically consists of the main ingredients

- CMU community,
- Agency,
- and agent community.

The so-called CMUs (Co-operative Manufacturing Units) provides meaningful functions to the manufacturing process. This definition is deliberately abstract and makes no assumptions about the physical realisation. In fact, there are three main types of CMUs:

- Manufacturing CMUs are used for the physical processing.
- Logical CMUs provide computational services. They are consulted by the agents for special tasks such as complex scheduling algorithms, database search, or the like.
- CMUs for special purposes provide necessary functions with respect to manufacturing process handling like transportation by TransportCMUs, administration of tools by ToolingCMUs, and supervisory control by SCADA-CMUs.

CMUs provide their services to the agent community with the aid of a Plug and Participate-Technology like Sun's Jini. This means that there is one (or more) lookup service(s) (LUS) that administrate the CMU-services and makes them available for the agent community. This allows a CMU to join the community without major changes in the system configuration and gives an agent the ability to easily find a CMU providing the service he is looking for. All these components are connected via a backbone network.

The agent community has an interface to the ERP system, the so-called Agency. The task of this component is the creation of the software agents and finally also their extinction upon completion of their task.

The agents scattered in this system are only partially mobile, depending on the necessities of their tasks. From a functional point of view, three agent types exist:

- Residential Agents (RA) are the interface between the CMUs and the agent community. They are stationary and tied to their specific CMU. Their task is to provide information about the capabilities of the CMU and to allow other agents to access the respective resources.
- Product Agents (PA) are associated with the actual work pieces being produced. They control the manufacturing process from the viewpoint of the individual product and take care of scheduling, resource allocation, or reporting. To this end, they have to be mobile.
- Plant Management Agents (PMA) finally organise the manufacturing process from a system-wide perspective. Their tasks include quality management, reporting, and the like. These agents are not necessarily mobile. If they are stationary, they reside in the Agency or on CMU for special purposes and perform their tasks by message exchange with the agent community.

The definition of the behaviour of these system parts (agents and CMUs) requires as basic technology for the implementation - a PnP-system as well as an Agent-system. To be independent from concrete technology abstraction layers were introduces. This means that as long as a set of basic requirements for a PnP-system is fulfilled, different technologies such as Jini, UPnP or Pini can be used. All physical system components that participate in the agent community, i.e., CMUs and the *Agency*, need an agent container or agent host providing a runtime environment for the agents to host its own interface agents as well as the product agents. This environment has to provide suitable communication facilities and an abstraction from the underlying operating system and hardware, as well as agent mobility. All these features where encapsulated in an agent abstraction layer, so that each agent system providing the requested functionality is easily adaptable.

Objective 1.3: Convergence towards an architecture that enables direct job control via an ERP system without MES and SCADA systems.

To fulfil this objective two aspects of the PABADIS-architecture are important: the realization of MES and SCADA functionalities by agents and the CMU-community and the realization of the intrinsic manufacturing process controlled by PAs.

[1] SCADA and MES in PABADIS:

In order to develop an architecture without a centralized MES component the PABADIS approach defines a system where the MES functions are implemented with agents, whose creation and objectives are committed to the *Agency*. SCADA functions require special attention due to their inherently centralised nature and real-time characteristics. For their implementation agents as well as a CMU for special purposes are required. The concept for the integration of MES and SCADA functions follows a two-part approach, which relies on a distinction between decentralised and centralised operations:

Hardware and material handling related functions are committed to agents, which need to negotiate with each other. PAs and RAs are jointly responsible for the physical realisation of products, starting from a drawn up manufacturing order up to its final completion. MES functions that result from agents' co-operation are resource allocation, routing, and detailed scheduling. This approach is focused on the product rather than on the process and enables a close insight and a straightforward access to work in progress (WIP) status and product tracking, allowing for the traceability of all components. The following table summarizes the different MES activities and shows how they are distributed among agents.

N°	MES and SCADA functionalities	PA	RA	PMA
1	Resource Allocation	✓̲	✓̲	
	Resource Status		✓	✓̲
2	Operations/Detail Scheduling	✓̲	✓	
3	Dispatching Production Units	✓̲	✓	
4	Document control	✓	✓	✓̲
5	Data collection and acquisition		✓	✓̲
6	Labour management		✓	✓̲
7	Quality management		✓	✓̲
8	Process management		✓	✓̲
9	Maintenance management		✓̲	✓̲
10	Product tracking and genealogy	✓̲		✓
11	Performance analysis		✓̲	✓̲

✓ Agent involved

✓̲ Main actor

Table 2: Distribution of MES-functions

Data management and information processing functions are performed by PMAs. These information agents are designed to collect data scattered over the network, and to present information in an appropriate way at a managing level. PMAs are concerned with process, quality and labour management, data or documents retrieval and forwarding. They provide indicators about plant processes with performance analysis functions.

Normally the access from the ERP level down to the agents' world is realized through the *Agency*. The *Agency* is the designated interface between the ERP system and the agents: it interprets manufacturing orders issued by ERP and creates Product Agents dedicated to the orders. PMAs are also created by the PABADIS *Agency* according to the production management needs, and hence they provide the ERP with all information about plant operations, resource use, and production performance. Among the above mentioned functionalities of a MES system, at least three of them are commonly performed in the industry by a particular tool called SCADA system:

- Data collection and acquisition
- Process management
- Performance analysis

However, a SCADA system is not well delimited, which means that SCADA software can perform more functions according to the respective vendors. In industry, currently existing SCADA systems are normally connected to each processing machine with an arbitrary network connection like a fieldbus connection in order to perform the data collection and acquisition function. One of the PABADIS objectives is to provide flexible production plant structures. This, of course, must also apply for the SCADA system configuration. The SCADA system must update its database and man machine interface in accordance with the production plant evolution. In order to meet this flexibility goal, the SCADA system shall not be directly connected to each processing machine but rather connected to the whole CMU community

With the use of a CMU for special purposes called SCADA CMU, the SCADA system is connected to the PABADIS community. Owing to this fact, it can be aware of any adding or removal of manufacturing CMUs which may happen in a PABADIS plant. Thus, it configures its database and its man machine interface in accordance with the CMU community. The Data collection and acquisition function is performed by directly using the PABADIS network.

[2] The manufacturing process in PABADIS:

The manufacturing process according to the PABADIS idea starts with the generation of a conventional manufacturing order by the ERP system to process a product. This order comprises the sequence of required processing steps together with the appropriate parameters for product processing. The manufacturing order is passed to the *Agency*, where it is translated into one or several product agents that join the multi-agent system.

Step by step, the product agent executes its production plan. The basic procedure it follows is always the same: it consults a lookup service, which is present in the network to find the CMUs that can provide the needed manufacturing service. The lookup service acts as a central service broker. Subsequently, the product agent contacts the residential agents of the CMUs and asks for information necessary to decide which service provider it should choose. Such information includes the availability (i.e., the earliest possible free time slot for processing), the expected duration of the action, but perhaps also the location of the CMU for the calculation of transportation costs. If necessary, the product agent will also contact entities providing special information with respect to necessary tools or other resources situated on CMU for special purposes. Based on these data, the agent selects the “optimal” CMU. The selection process sketched here is fairly simple. More complex scheduling procedures would involve the communication of the agent with competitors to dynamically change the resource allocation and create different sequencing and dispatching plans. Although the development of distributed scheduling algorithms is not in the focus of the PABADIS project, such advanced features can be added to the agents at any time by replacing their selection and negotiation modules at creation time. Alternatively, specialised logical CMUs or CMU for special purposes can provide these functions.

Throughout the manufacturing process, the product agent guides the work piece. Upon completion, it returns to the *Agency* and is terminated there. The *Agency* then generates a report to the ERP system using the data the agent has collected on its way through the production (if so desired by the ERP system in the production order). In parallel, plant management agents are created by the *Agency* to fulfil specific control or supervision tasks that are not related to individual products.

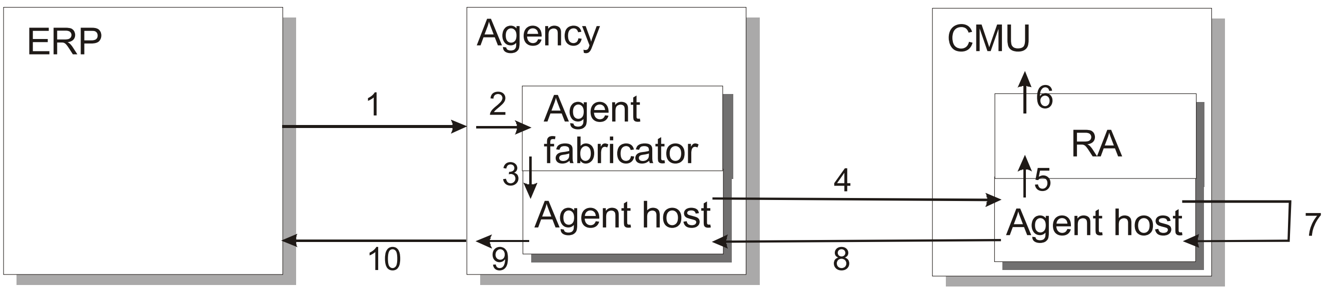

Figure 1: Steps of a life cycle of a product processing in the PABADIS-system

(1) The ERP system produces a manufacturing order and sends it to the *Agency*

(2) This manufacturing order is transformed to a work order containing several tasks by the *Agency* and handed over to the agent fabricator to create a PA.

(3) The agent fabricator creates the PA in the agent host and these agents are initialised with work order description.

(4) After the creation, the PA migrates through the network in order to perform the tasks described in the work order. To do so, it travels to the first CMU and resides there in the agent host.

(5) Within the CMU the PA accesses the RA to be processed.

(6) This access is passed to the function by RA in a fix way.

(7) When the task on the CMU was performed, the PA migrates to the next desired CMU to process the next task (5,6).

(8) After finishing the whole work order, the PA returns to the agent host of the *Agency* to deliver all collected data,

(9) to signalise the finishing of the work order and to terminate.

(10) The *Agency* will deliver the whole PA data to the ERP, in order to enable the ERP to do its work.

Objective 1.4: Design an adequate platform for mobile and residential software agents.

A central point in an agent-based system like PABADIS is the design of an appropriate platform as environment for mobile and stationary agents giving single agents the possibility to fulfil their specific tasks within the system. For the PABADIS-system the aim was to define a platform which is as generic as possible. This requires independency from the used agent-technology and the possibility to be executed on different hardware platforms. To solve the first problem an agent abstraction layer similar to that for the PnP-system was introduced. Upon this layer all PABADIS-specific functionality was defined. This enables as long as the used agent system fulfils a basic set of requirements (support of mobility, JAVA-api) an easy adaptation on different technologies. The platform independency is given by the usage of Java as programming language for all generic PABADIS-software. All agents in PABADIS follow the same basic structure:

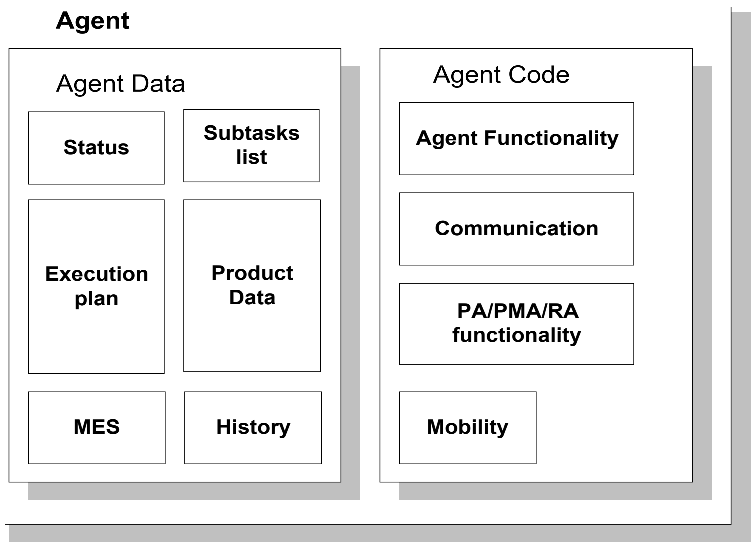

Figure 2: General structure of a PABADIS-agent

The methods and the data are encapsulated by the object and are only accessible via defined interfaces. Every agent consists of two main components: code and data. The code of an agent includes standard methods, which are the same for every agent, and specific methods, which concern the concrete task of the agent. As described above there are three agent types in the PABADIS system:

- Product agents (PA) : A PA is a mobile agent, which is responsible for manufacturing a product

- Residential agents (RA): A RA is an agent, which provides the interface between the PA and the CMU.
- Plant Management Agent (PMA): A PMA is an agent, which provides an interface between the agent community and the ERP and SCADA systems.

There are two subclasses of the “Agent” class: “Product Agent” and “Stationary Agent”. The latter itself also has two subclasses, namely “PMA” and “Residential Agent”. The “Product Agent” class implements necessary features of the PA (the most of them is mobility) and the “Stationary Agent” class is used for PMA and RA creation.

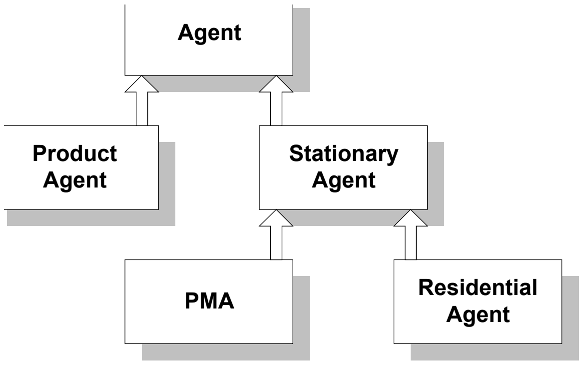

Figure 3: PABADIS agent class hierarchy

The “intelligence” of the system is mostly contained within the PAs. The RA and PMA are mainly interfaces to the PA community. The RAs represent the CMUs on which they reside, but the only “intellect” they have, are the necessary skills for accessing the underlying CMU production resources and for an appropriate scheduling of the resource allocation and negotiation with PA. The same argument applies to the PMAs, but in contrast to the RAs the PMAs normally represent an interface to the ERP system.

Objective 1.5: Definition of possible strategies and functions of a single agent which executes a special job.

As an example for the most sophisticated agent-type in a PABADIS-system the functions and strategies of a Product Agent (PA) shall be described here. A PA is one of the key components of the PABADIS MAS because it provides the intelligence of the system. The main task of a PA is to perform a work order, provided by the *Agency*. Due to the fact that the PA is a temporary unit in the PABADIS system, the PA life cycle can be described in the set of consequential actions which the PA has to pass during its life. Generally speaking, there are six main steps of the PA:

- Creation
- WOparsing
- Resource allocation
- Migration
- Task execution
- Termination

Additionally to this, there is the initialisation step. This procedure is more or less a sub-step of the creation when the basic installation actions have to be done.

During the creation step a PA receives its work order, set-up variables, recognises its place in the community and prepares the work order to be analysed. Within the creation process the PA is a passive element, it does not perform anything, except during the initialisation step, which is a part of the creation process. The creation state means that a PA instance is created, initialised and ready to start the execution of the work order. It is finished with completing the initialisation procedure.

Subsequently, the PA starts working with the work order while it is still located in the Agent Container of the *Agency* and performs all necessary communications remotely.

Within the next step, the work order parsing, a PA parses the work order to construct an execution plan. To do this, the PA analyses all branches of the work order, which are represented in a graph view, and creates a string array with a sequence of actions, which will be used by the PA for resource allocation. It analyses a work order by parsing the graph and creates an execution plan. As a result all paths are defined and ready for resource allocation. Each path represents a set of tasks which have to be executed in order to create a product. Paths depend on the availability of machines which can execute the functionality needed by the PA. As the result of this process for every function a certain CMU or a set of CMUs are assigned and thus, the PA knows addresses to go to.

The PABADIS system operates on products and resources. A PA contains all data concerning necessary tasks and operations that have to be done in order to produce a product. In contrast, a RA contains information about the available resources in a single CMU. Both the PA community and the RAs perform the scheduling procedure in order to allocate a resource. During the resource allocation step the path of the work order execution is chosen. Each function is assigned to only one CMU. The PA tries to reserve a timeslot in the RA schedule, which represents the usage of the CMU's function. Start of execution and finish of execution is known. RAs/CMUs are allocated and thus, the respective resources are reserved. Realizing this behaviour, the PA can choose the best CMUs for his job by analysing different parameters by different criteria such as fastest executing, cheapest cost of task execution and so on. This also includes a negotiation process between PAs where priorities and PA-synchronization aspects play an important role. Basically, the PA uses two scheduling parameters to adapt the behaviour on different requirements of the plant:

- Depth: the depth of the schedule, means, how many steps are scheduled in advance
- Window: how many steps have to be calculated in one step as a bundle. The window size is always equal or less to the depth of schedule.

When the resource for a task is allocated, the PA migrates to the Agent container of the CMU, which performs the task.

During this migration-step a PA migrates to the CMU, checks its location and waits for an execution. This step is mostly due to an agent platform (currently Grasshopper). A PA initiates a migration, but does not control the process. Migration is done by agent hosts of the appropriate CMUs (current location of an agent and a destination CMU). Within this procedure the PA prepare the migration by checking the available memory on the destination CMU. This is done before the PA is moved. Furthermore, within the whole migration process a PA is backed up by a so-called "Shadow mechanism" that enables a restoring of lost PAs in order to keep the robustness requirements of the PA.

During the task execution a PA just waits for the end of execution. A RA informs a PA about a result of work piece processing. Hence, a PA is just a passive element during a task performing. There is a possibility to interrupt a process, but only in critical situations. The exact mechanism depends on the scheduling mechanism. But usually the PA cannot interrupt or stop the process. Within this process the RA is the active partner, which communicates with the CMU via a so called Function Control Module (FCM).

The termination of a PA is the process where all components of an agent are removed from the system. Code and data are logically and, if necessary, physically deleted. This depends on the concrete agent platform mechanisms and hardware requirements of the system. The termination of a PA happens in two cases:

- Work order completed: In this case the PA moves to the *Agency* and "dies" there. Hence, the PA initiates its removing from the system by itself
- Impossibility to complete the work order

To realize the overall process of a work order execution the PA is able to use a set of basic mechanisms enabling to make “intelligent” decisions as reaction on the current state of the plant. This includes:

- a PA priority changing mechanism
- Routing mechanism for the transportation tasks
- production time estimation

All this enables a PA to efficiently execute a work order in a PABADIS environment.

4.2 Development

Objective 2.1 Develop the plug-and-participate concept and test its technical feasibility for machines and systems. Develop a bottom-up conceptualisation for functions of industrial machines that will be enabled to interact with each other and the company's ERP system using plug-and-participating concepts.

By the term “plug-and-participate”, we primarily mean the self-organisation of services within the network from an application point of view, i.e., communicating partners can use each others services without a manual configuration of the respective interfaces. Independent of this aspect is the low-level network setup for the individual devices, which is frequently subsumed under the term “plug-and-play”, but not necessarily the primary goal of PABADIS.

At any rate, plug-and-participate as we understand it requires some sort of “middle ware” layer enabling the abstract formulation of distributed objects and services. For PABADIS, Jini was selected as the middle ware of choice. The reason for this decision was a rather pragmatic one. The agent systems considered are based on Java to permit mobility. Hence, Jini snugly fits in and complements the system. But of course Jini can be replaced by any system providing the same functionality. What is actually used within the frame-work of PABADIS is Jini's lookup service. A new CMU (to be precise, its residential agent) that is attached to the system registers with the lookup service and announces the functions it can provide. The product agents in turn use the lookup service to find appropriate CMUs as described in the previous section (see also Figure 4).

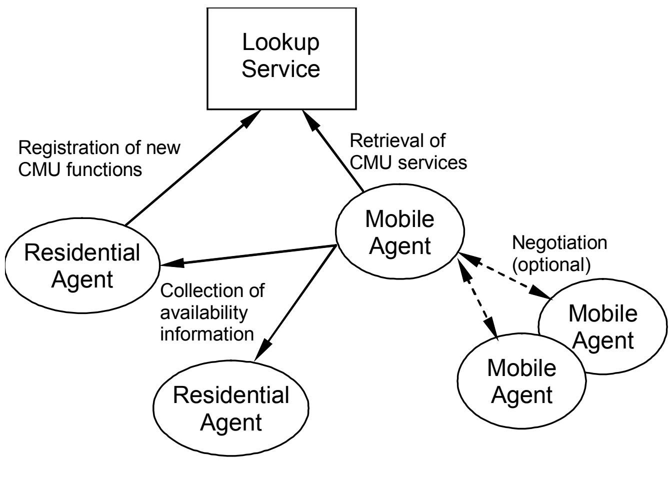

Figure 4: Communication relations between the lookup service and the various software agents

It is noteworthy that many pure multi-agent systems also have some sort of lookup service. This procedure is usually performed by some special facilitator-agent or broker-agent. However, there is no self-registration in these systems, and this is why we use Jini. At any rate, the PABADIS concept is modular enough to replace the technology used for the lookup service by virtually any other one at a later date.

Mapping the requirements of a PABADIS system to features of Jini gives an overview how the PnP-system of PABADIS works. This, of course, expects a mapping of the Jini components to the PABADIS components. In this way, the relationships of the Jini components are automatically applied to the PABADIS infrastructure:

Jini component	**PABADIS counterpart**
Services	Functions of the CMUs
Service users	Agent community, which uses the CMU features via the RAs of CMUs
Lookup Service	Lookup Service

This mapping already shows the actions needed to be performed in order to establish an ad-hoc network of CMU functions via the Jini technology, which can be used by the agent community:

- performing the discovery protocol, done by CMUs as well as by the PABADIS-*Agency*
- performing the join protocol, done by CMUs and the PABADIS-*Agency*
- performing the lookup protocol, done by agents via the CMUs, also performed by the *Agency* via its P&P module
- performing the lease protocol, done by CMUs as well as by the PABADIS-*Agency*
- creation of attributes from the CapabilityDescription (CD), done by the CMUs
- registration for remote events, done by CMUs, agents as well as PABADIS-*Agency*

Since the Jini technology is in some cases to resource expensive within the project a Jini like P6P technology named Pini was developed enabling the application of P6P features also on limited devices.

Objective 2.2 Design generic platforms to enable the communication with mobile agents which can be used within any future PLC, CNC and RNC (vendor independent).

A Co-operative Manufacturing Unit (CMU) is one of the basic components of a PABADIS system. It provides all necessary features to the agent community in order to enable the agents to fulfil their tasks. This either means to fabricate a product or to retrieve information about states, load, etc. from the CMUs. Furthermore, these CMUs are connected via a network to each other. Hence, the agents can either migrate directly to the CMUs (in case of PAs) or contact the CMUs from remote (in case of both PAs and PMAs). In detail a CMU is characterized as follows:

- A CMU is a PABADIS entity comprising one or more function(s) which provide(s) manufacturing production capabilities, a set of common support services to exploit the PABADIS environment and an agent host. It includes any needed hard- and software resources to perform its function and services.
- A CMU has the ability for self-integration into a PABADIS system via a P&P technology.
- PAs and/or PMAs can use a CMU's function(s) and common services via a well defined interface of the residential agent in order to perform their processing tasks.

As a consequence of this definition, the functional model of a CMU consists at least of the subsequently listed components:

- Control module,
- P&P module,
- Residential agent.

For structural reasons, a component, the so called common feature module, was introduced, which contains all modules that all CMUs have in common. Generally speaking, the components, which are part of this module always have the same implementation. Hence, this module can be taken from one CMU and directly be installed on another CMU. This module is a basic part of every CMU. It summarises several sub-modules, which have been described above and are listed subsequently:

- P&P module,
- MES module,
- SCADA module,
- HTTP server,
- XML parser,
- Authentication module (for security reasons).

The following figure shows the general implementation structure of a CMU.

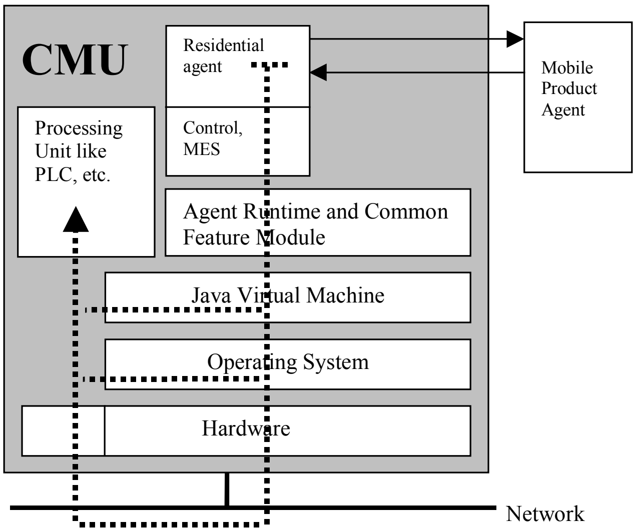

Figure 5: General implementation structure of a CMU

As it can be seen in the figure above, there are interfaces necessary between the shown components. For the integration of a CMU in the Agent community the RAACI (RA-to-Agent-Community Interface) and for communication between the RA and the (automation) functions the RAFI (RA-to-Function Interface) were defined.

Depending on the basic hard- and software of the device implementing the CMU it can be distinguished between different types of CMUs. These differences between the CMU types can be seen in the different types of access to the functions provided by the CMU. These provided functions are included in the processing unit component as shown in the figure above.

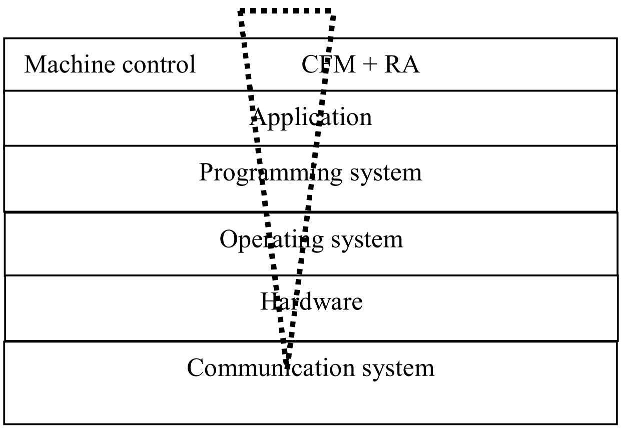

Figure 6: Distribution of CMU parts

Scientifically speaking, the different types are based upon the separation of machine control on the one hand, and the Common Feature Module and the residential agent on the other hand. The figure above shows this separation in a schematic fashion. There exist two extreme possibilities. The first extreme is the weak separation, which means that the requirements of the machine control and the CFM/RA must be combined. Moreover, this type of separation leads to the first CMU type, the so-called *Java-based CMU*. It provides access to the functions provided by the processing unit directly via the Java Virtual Machine. A very important requirement for this CMU type is its real-time capability. This means, the JVM must be real-time enabled, because the control software has to consider appropriate real-time constraints.

The other extreme is the *COM-linked CMU*, which shows a strong separation of machine control and Common Feature Module and RA. Here the access to the processing unit happens via the hardware and an additional network connection. This additional network connection can either be realised via an Ethernet connection or a fieldbus system. Examples of this type of CMU are a PLC combined with a standard PC providing a Java environment for the Common Feature Module and the corresponding agent environment. A special kind of a COM-linked CMU is the Java-linked CMU which is characterized by the overall usage of Java as implementation language (for generic and function part of the CMU). This enables (in contrast to normal COM-linked CMUs) to use the same implementation paradigms on both sides resulting in an efficient and easily reusable implementation.

The third CMU type is the *OS-linked CMU*. In this case the access to the functions is via the OS of the device. Of course, there only a moderate separation of the machine control and the Common Feature Module and the RA can be stated. To be more precise, the separation happens only on the upper layers of the operating system, the lower layers are for all CMUs the same. A typical realization of such a CMU is a control device with a RT-operating system which provides an additional Non-RT-JVM to run the generic part of the CMU-software (CFM and agent-system components).

To guarantee the applicability of PABADIS-CMUs for a wide range of control devices the access to the (automation) function from the agent world is a critical point. Hence an interface is needed which defines a standardized communication between both parts, but also enables a wide range of usable devices without requiring a high implementation and adaptation effort. Therefore the Residential-Agent-to-Function-Interface (RAFI) was defined. The RAFI ensures a generic representation of the CMU functions to the PABADIS agent community. Specifically, it decouples the generic PABADIS world and the (in general non-generic) function world. Moreover, it decreases the complexity of the PABADIS entities and consequently, it increases the flexibility of the whole system in providing a standardized access path and view to all functions available in a PABADIS community. This increased flexibility is achieved by four main components:

- Capability Description: It provides a standardized way to describe the functions of a CMU in a detailed but generic fashion. The representation/structure of content of the CD is derived from the Process Specification Language (PSL, see [PSL]). Strictly speaking, the CD is strongly related to the PSL ontology and additionally, it extends the definitions given by PSL. Furthermore, the CD is XML-formatted and, thus, in a generic format.
- Function Control Module: The Function Control Module (defining a finite state machine), is used CMU internally to control the function execution steps in a CMU. Specifically, there is one generic FCM in every CMU. State transitions are triggered by the RA and the function itself. Consequently, this module ensures the generic access path to the specific functions of the CMU.
- Parameter Interface: Before a function can be used for product processing, the function needs to be parameterised. Therefore, this interface provides the generic path for setting the specific parameters and, hence, provides read/write access to the parameter values of CMU function(s) for the agent community. Moreover, this interface is in close relation to the CD and also to the FCM.
- IRI: The Information Request Interface (IRI) is used for accessing device data, independent from any status of the FCM. For example, the SCADA system, PAs, PMAs, etc. may need information regarding the status of the function or the device. Therefore the appropriate entities request the IRI in order to retrieve the required data.

Additionally, the RAFI consists of a further component, which controls the exchange of application specific events: the Event Control Module. In general, it is used for exchanging events not directly related to the function control, but to the function itself. This means, via this module an application specific information flow can be stated. For example, events for requesting information from the function or the RA, for requesting quality data, tracking data, status of the function, etc. are exchanged. Here it has always to be taken into account that the destination of such a request is a real-time area; specifically, the event's information flow must not have any impact on the function's real-time constraint control flow. Therefore, the direct access to the function must be avoided.

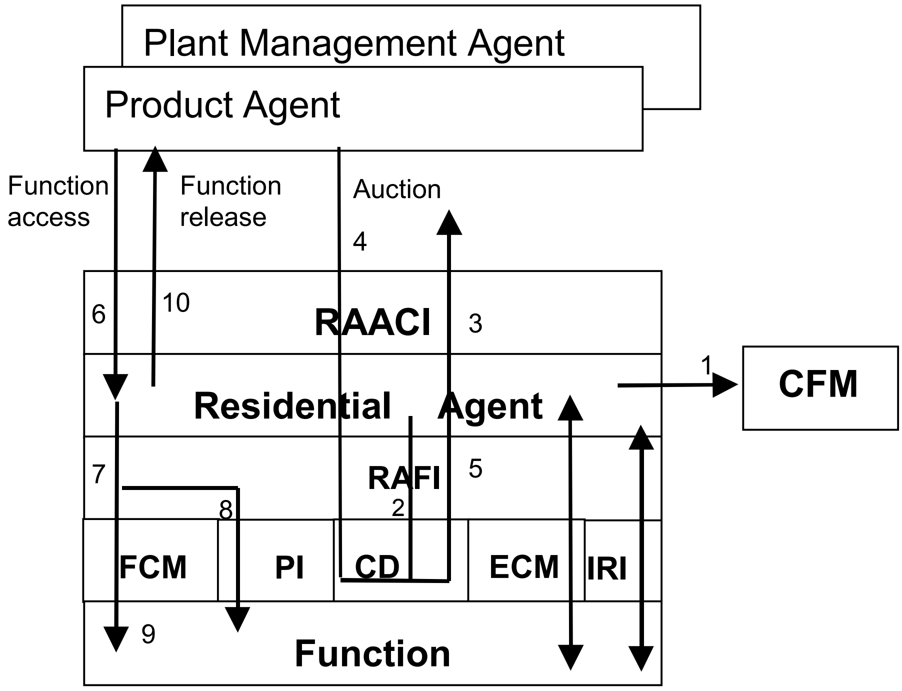

Figure 7: The functionality of the RAFI within the PABADIS system

Objective 2.3 Develop generic platforms which can be used with any future ERP system to enable the flexible creation of job controlling mobile agents (vendor independent).

The developed generic interaction platform between PABADIS system and ERP system is based on the two main building blocks Agency and Generic Interface to the ERP system. The later one is from the final implementation a part of the agency but logically it is separated.

[1] *The Generic ERP-Interface*

During the past few years, ERP systems have opened up their tightly interwoven modules and created application programming interfaces (APIs) to connect to 3rd-party best-of-breed systems. ERP systems are offering a broad range of open integration schemes, including extensible markup language (XML) messaging and proprietary connectors or open APIs, since easy integration to 3rd-party applications has become a key point for them. Nowadays, the industry faces significant interoperability issues as it seeks to provide solutions for distributed systems consisting of clients and servers of heterogeneous hosts to enable joint service operations. XML and related technologies offer promise for applying data management technology to documents and, also, for providing a neutral syntax for interoperability among disparate systems.

The main objective, when developing an interface is to provide a truly interoperable technology, which can be seen as a stand-alone product, able to maximise the value and reuse potential of information under its control and provide to the internal systems of a company the ability to work with other systems without special effort.

Following the above approach, we try to proceed to the creation of a generic interface in PABADIS, which has the capability of connecting with different ERP systems, in spite of some difficulties regarding the general ERP architecture, consisting of accessing both to the application and to the database server :

- ERP system's application , has the objective of integrating as much as possible all the activities of an enterprise, including i.e. human factors, supply chain, etc. This global integration theoretically suppresses all interfaces and additional off-line processes, so that different users can share at the same time the same set of data. Such an integrated system, combined with a huge quantity of data, processes and functions, renders the ERP system difficult to interface easily.
- ERP system's database server has to maintain a permanently, quasi real-time database, and warrant data security and integrity, but access to database structure is strictly controlled, so that a tailor-made connection with PABADIS is not straightforward.

Facing these difficulties of designing a generic interface integration of the PABADIS MES system in relation to an ERP system, we presented a stepped approach, based on the following considerations:

- XML, which is a well known interoperability technology, is used to structure data when transferring data to and from the ERP system. This technology is maintained by the W3C consortium, and tends to supersede all former classical EDI systems (the example of SAP's Business Connector is significant). In PABADIS, the same format is also used internally to communicate with agents.
- Data transaction will be ensured by means of an additional database repository. In order to provide a highly flexible, ERP-adaptive interface, it is proposed to map the production data of interest onto this additional database, which will serve as a buffer between the ERP system and the PABADIS community.

The architecture of the generic ERP interface is described in Figure 8. It is composed of a generic interface and some specific connectors, each of them connects to a particular ERP system.

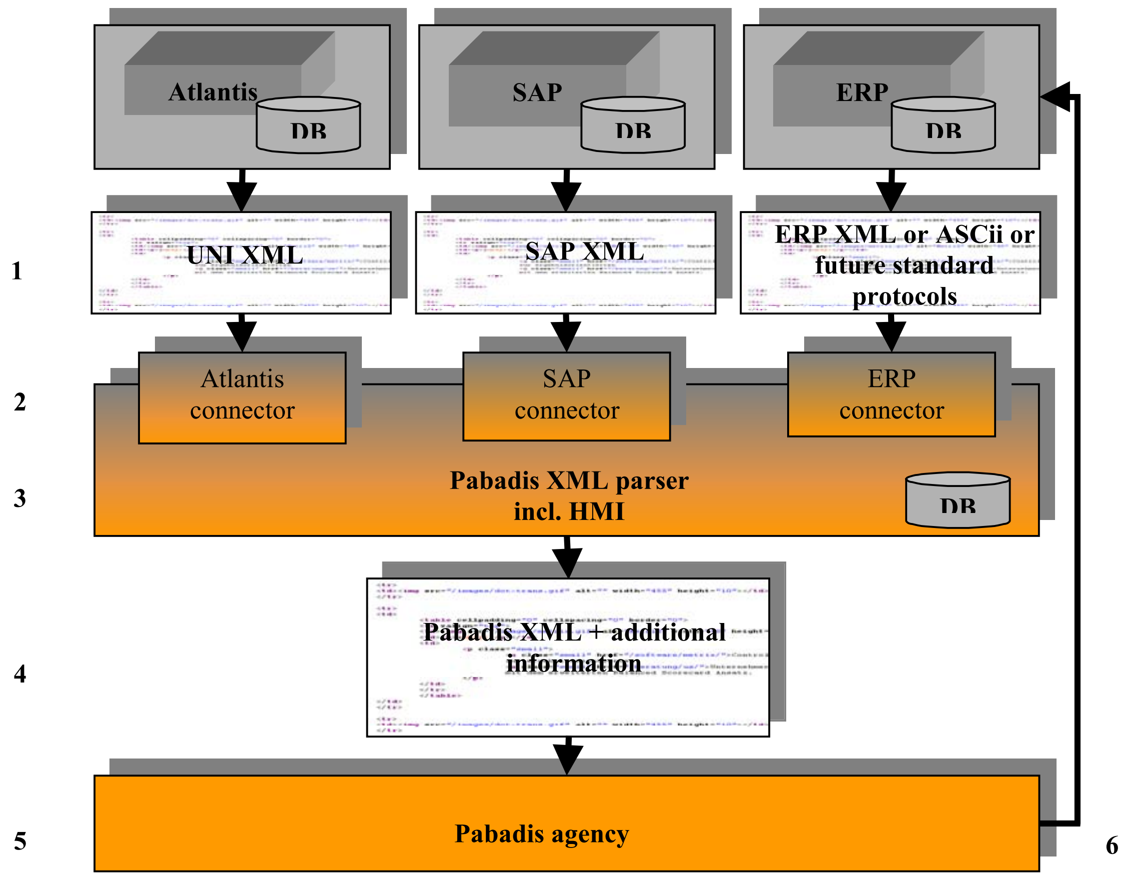

Figure 8: Generic ERP-PABADIS Interface

The role of the different layers is defined as follows :

1. The ERP system generates its own proprietary XML file containing information about product, manufacturing process, material etc.
2. 2. The XML parser gathers and filters the ERP XML file using a ERP customized data connector.
3. The user can add additional (still missing) information using a HMI. Furthermore additional information could be added from a database containing information about e.g. PDA automatically.
4. The XML parser finally generates a „standard“ Pabadis XML file containing all necessary information about the product manufacturing which is sent to the Pabadis Agency.
5. The Pabadis Agency handles the Pabadis XML file.
6. The Pabadis Agency gives feedback to the ERP system (e.g. monitoring)

The Agency-ERP Interface waits for messages coming from and going to the ERP. These Data use an XML representation, their structure follows the DTD format. After having parsed and analysed the XML data issued from the ERP, the Agency is in turn in charge of feeding corresponding Product Agents with proper Work Orders.

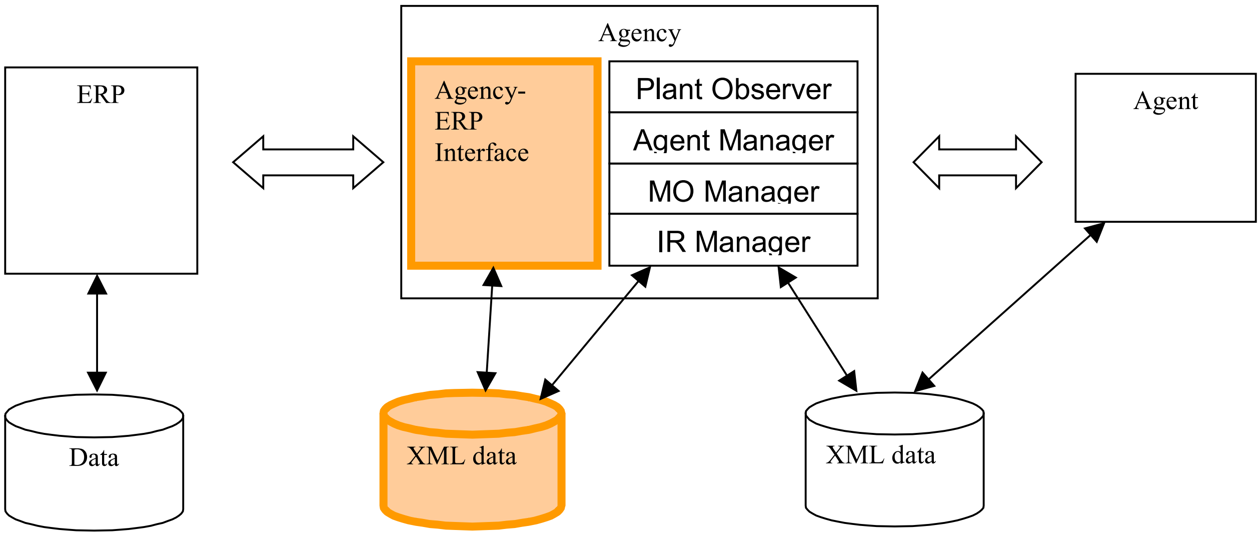

Figure 9: Structure of the ERP – Agency interface

[2] The Agency

The fabrication of agents is the main task of the Agency. Concerned agents are PAs and PMAs. The Agency architecture in Figure 10 shows the overall composition of the Agency. Modules are differentiated between static and dynamic elements.

Static elements :

- Interface to ERP : this module represents in fact the communication link with ERP, on the Agency side. It is limited to the access methods to the Agency database.
- MOManager, IRManager, PSobserver have exactly the same behavior : each time a new request is delivered, the corresponding module creates a specific Plant Management Agent, get its report, and finally deletes it.
- Agentfabricator is a module that offers agent creation methods
- AgencyRA is the Agency Residential Agent, created at the start of the Agency.

Dynamic elements :

- PMAs are dynamically created in the Agency : a MOPMA for a Manufacturing Order, an IRPMA for a Information Request, a PSPMA for a Plant Structure request. In addition, a MOPMA will create the Product Agents that will be in charge of the Work Orders.

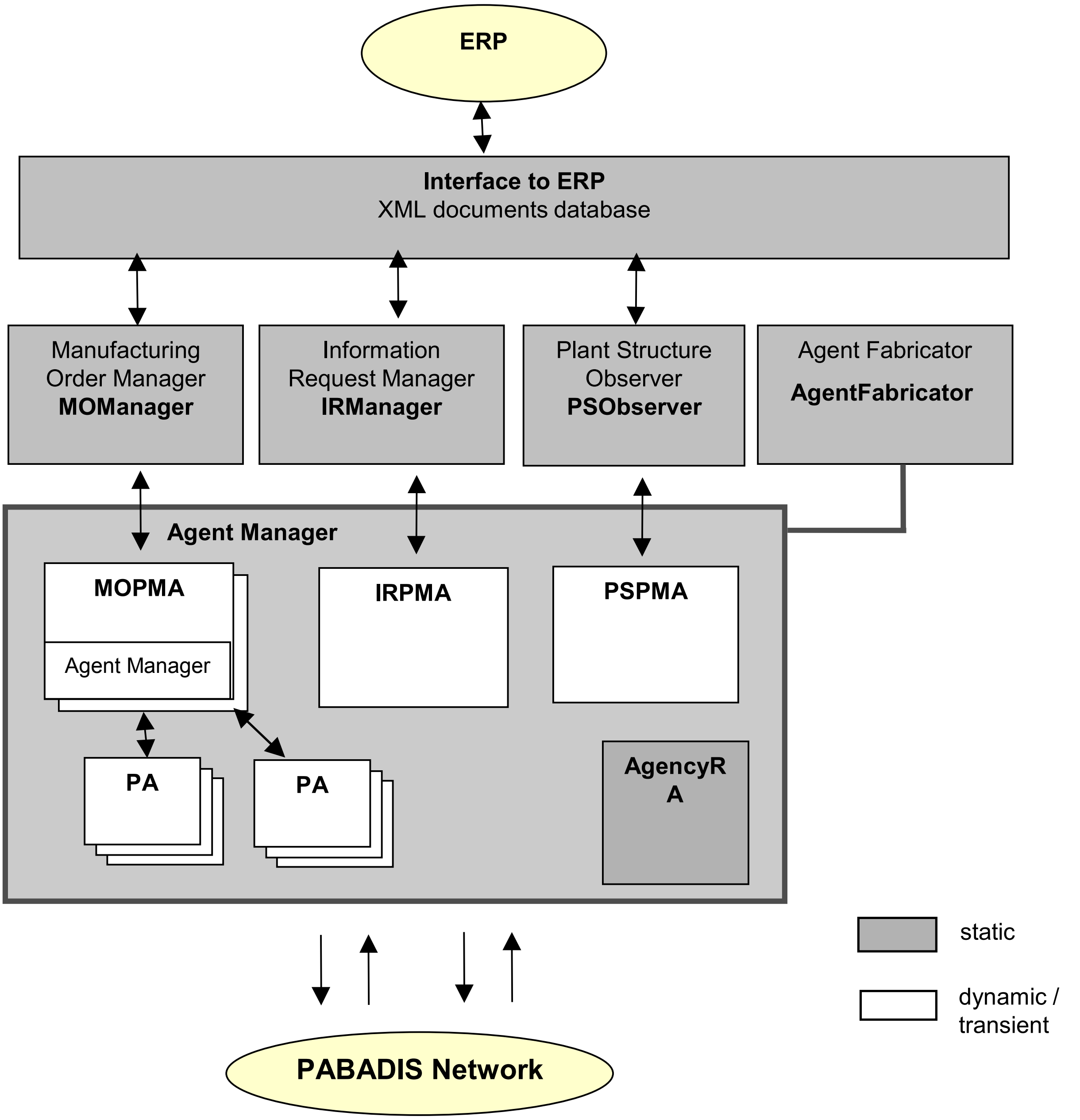

Figure 10: Agency architecture

The Dialog with ERP through the ERP interface splits the data exchange into 3 types :

- infrastructure data, handled by the Plant Structure Observer,
- manufacturing data, handled by the Work Order Manager,
- other data, handled by the Information Request Manager.

The creation of product agents is based on the information received from the ERP system given by manufacturing orders. A manufacturing order comprises the sequence of required processing steps together with the appropriate parameters. The production order is passed to the Agency, where it is translated into a PA and joins the multi-agent system. Throughout the manufacturing process, the PA guides the work. Upon completion, it returns to the Agency and is terminated there. The Agency then generates a report to the ERP system, using data the agent has collected on its way through the production (if so desired by the ERP system in the production order).

A PA is a mobile agent, created on the request of the ERP system for launching a production. The ERP system produces a manufacturing order, which has to be decomposed by the Agency into elementary work orders, each of them being taken in charge by a specific PA.

The data associated with a manufacturing order are issued from a preliminary planning; in addition to the data given above, they also may incorporate data from a PDM (process data management):

- Product designation
- Quantity
- Start date
- Due date
- Bill of material (BOM)
- Product data (formulae / recipes / drawings or part programs)
- Material/parts availability

In the case when products are made with blended operations like assembly, disassembly, scrap, and recycling, the Agency creates as many PAs as necessary and combines the different issues.

The Agency provides product agents with the execution plan of the whole manufacturing order, so that synchronisation and co-ordination can be managed by the agents themselves. At completion of the elementary order (work order), the PA reports to the Agency and is then deleted.

The **Plant Structure Observer** is that part of the Agency which informs the ERP about the updated plant configuration. Exploiting the Plug-and-Participate functionality, it provides the actual plant structure on ERP request or on plant configuration changes, at a level and a frequency compatible with ERP requirements. That means, the "granularity" of information, its nature and details, the periodicity of updates or events will be fixed by the user, depending on the specific features of the manufacturing process, and on the monitoring requirements at the ERP level.

A central role in the execution of work orders plays the **Work Order Manager**. This module processes the Manufacturing Orders issued by the ERP. Its activities are :

- MO setting and WO generation: Once the complete Manufacturing Order has been elaborated, the WO Manager generates as many Work Orders as necessary, transmits them to the Agent Manager, which will associate them to Product Agents.

- MO Monitoring: During the execution of a MO, the WO Manager :

  1) gathers reports delivered by PAs, mainly at the end of the related Work Orders.

  2) checks the Work Orders and compares them with the initial planning. The Agency may have here the possibility to compensate minor production problems by creating complementary Work Orders.
  3) in case of complete Manufacturing Order, or in case of major problem, a report is sent to the ERP system. If production adjustment is needed, a new WO is created, and a warning is sent to ERP.
  4) when requested by the ERP, the WO Manager returns the current status of a MO.

During execution, the ERP has the capability to control a released MO: cancel, suspend, resume.

The objective of the **Information Request Manager** module is to answer to all other information needs issued from ERP or a CMU. Parts of MES functions will be implemented in this module :

- Document control
- Data collection and acquisition
- Maintenance management
- Performance analysis

Except for document control, the IR Manager will require the creation of a dedicated PMA.

The **Agent Manager** permits:

On request from the WO Manager

- to generate PAs responsible of the execution of a WOs,
- to control the state of PAs (delete, suspend, resume, find, access)
- On request from the IR Manager,
- to generate PMAs dedicated to the request
- to control the state of PMAs (delete, suspend, resume, find, access)

The Agent Fabricator is part of the Agent Manager, it is connected to the Agent Host, it creates PAs and PMAs and provides them with a mission and related data.

Transient entities within the Agent Manager are:

<u>MOPMA</u>, it is created with a list of WOs :

- it creates in turn a PA for each WO
- it maintains a PA reference table (unique agent identifiers), a WO table, a WO report table
- when MO is completed, MOPMA is deleted by MOManager
- In addition, MOPMA disposes of agent management methods (from AgentManager) for the control of the PAs.

<u>IRPMA</u>

- sends requests to CMUs and waits for the reports

<u>PSPMA</u>

- sends a request to the LUS (Look Up Service) and waits for the report

Objective 2.4 Program generic software platforms for mobile agents in production plants. In this way, standardisation of metadata for component (object) design and reuse may be facilitated for independent industrial IT vendors; according to this pattern, a component model specifies agent interfaces and describes the design, assembly, and deployment of those components.

An important goal for the developed PABADIS-system was to enable an easy applicability to different systems based on standardized IT technologies. This leads to the design decisions to:

- define a generic part for all needed PABADIS-components which can be easily adopted on different control architectures using defined interfaces to the control level
- use system wide XML as common representation for all information such as Manufacturing Orders (MO), Work Orders (WO) and Capability Descriptions (CD).
- define interface layers for all used basic technologies such as the agent- and PnP-system, which enable an easy exchangeability of the underlying technologies. (currently Jini is used, but can be replaced without much effort by e.g. Pini or UPnP as well as the used Grasshopper can be substituted by other agent systems).

As a result the practical implementation of a PABADIS-system is (based on the reference-implementation) mainly a configuration and creation of XML-descriptions of system elements. To sum up, a vendor must fulfil the following requirements in order to PABADIS-enable a plant:

1. Computers and devices must have installed Java 2 (JRE) and a PnP-system (e.g. the Jini classes/interfaces).
2. Computers and devices must be network-enabled (in most cases, IP, TCP, UDP).
3. The necessary classes for the CMU configuration and the RAFI must be available (non generic CMU-part).
4. An Agency instance is necessary as the interface to the ERP system of the plant.
5. Capability descriptions for CMUs are to be made available.

Finally the consortium has developed a new common generic product and process description languages which can be used within nearly all industries to describe products and production processes and to compare them with respect to the practical use-ability of as production process to manufacture a special product.

4.3 Testing

Objective 3.1 Realise a test case for the developed platforms within a real industrial surrounding.

The main goal of the implemented demonstrators is to show the applicability of the PABADIS ideas in industrial practice. Hence, the following main features of a PABADIS plant should be represented:

- Decentralized intelligence
- Agent and service based production execution
- Modularity of all plant components

In this context it was envisioned to proof the technical functionality of the developed infrastructure concepts and software in a real plant scenario and to show the applicability of the PABADIS concept for different manufacturing processes. Due to the complexity of the developed software and the number of different components to be integrated two demonstrators were planed. The first one, based on a small plant model, serves the integration of components and tests regarding complex PA behavior, such as different scheduling strategies. In contrast to that the second demonstrator is based on real industrial equipment, the so called “Lernfabrik” (experimental plant), and should serve as test bed with more complex, industry alike constraints which mainly result from the automation system.

The first demonstrator is based on a “Fischertechnik”- plant model. This model consists of 3 machines, 8 turntables, and 10 conveyors. The machines are equipped with three different tools placed on a turret for automatic tool changing. Logically the system can be seen as a paint shop with three multipurpose machines. Within this paint shop raw cubes will be cleaned, primed and painted with black or white lacquer.

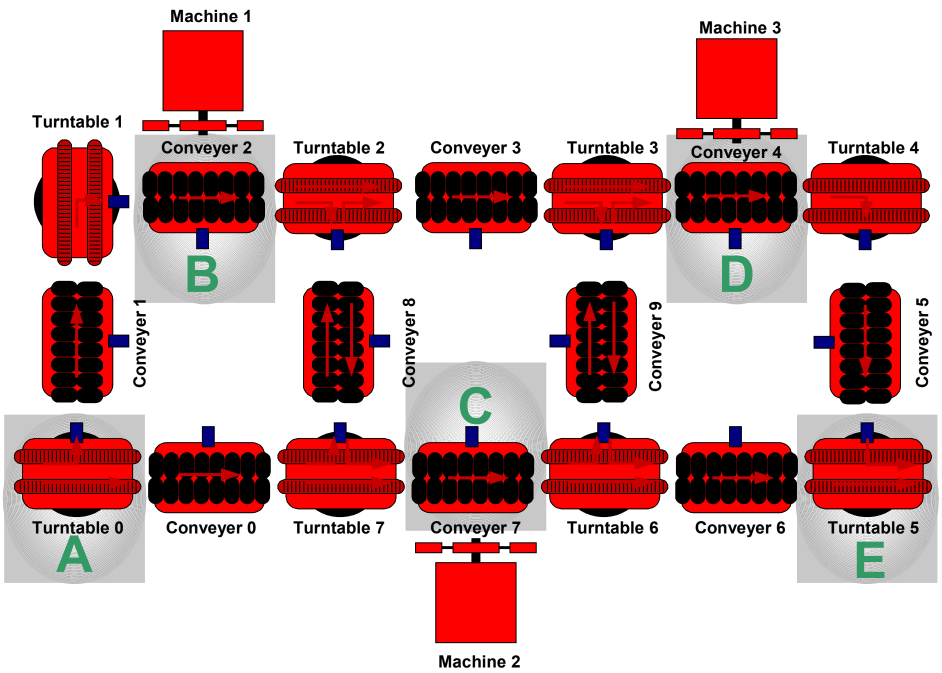

Figure 11: Structure of the Fischertechnik-model demonstrator

With respect to the demonstration aims the demonstrator is divided in 4 modules, each of them controlled by one PLC. These modularisation is equal to the set of CMU within the PABADIS model system. Each machine is controlled by a JetControl based CMU, while the transfer system is controlled by the Java-linked TimeSys CMU.

The second demonstrator (experimental plant) is based on facilities which are built out of standard industrial parts and originally were designed as training facility for students. The plant consists of 1 high rack storage with a capacity of 60 pallets, a rack feeder for taking pallets out of the storage (and for putting them in again), a transfer system to transport pallets to the assembly places, a manual work place for manually assembly and finally an assembly robot. To realize a safe operation several constraints in the interaction of transport system and other entities have to be considered.

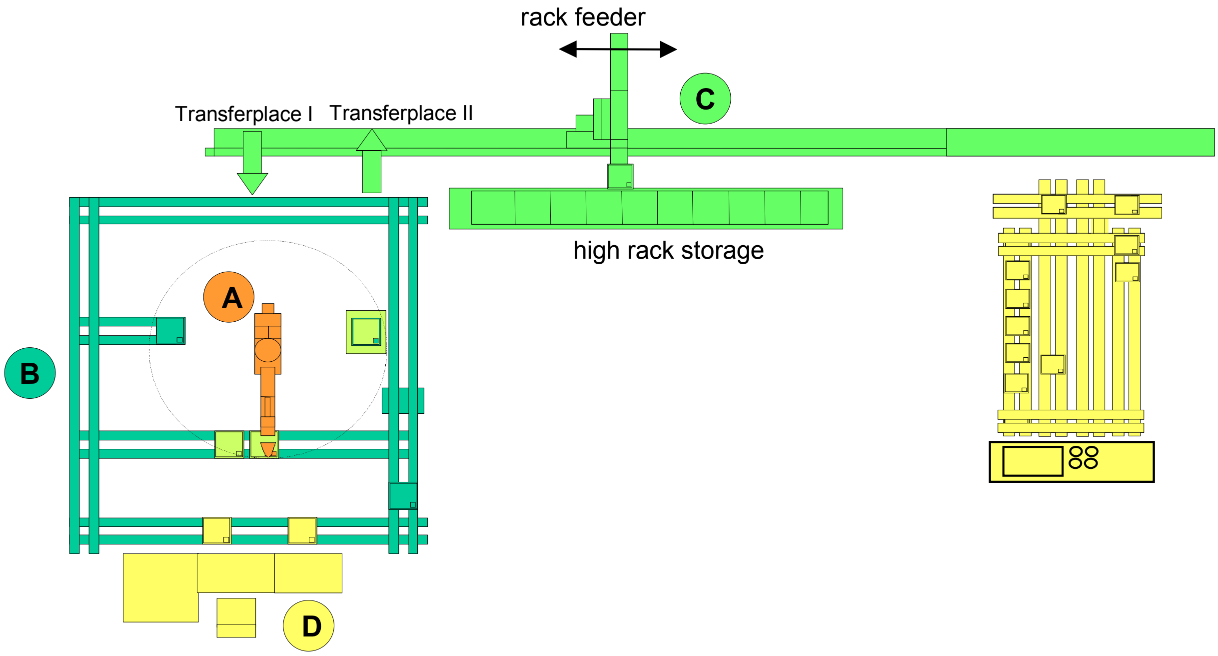

Figure 12: Structure of the experimental plant

For the implementation of the PABADIS-System the following modularization of the system was defined:

- *robot (A)*
  The robot is controlled by a Bosch robot control and a Phoenix SlotPLC.
- *transport system (B)*
  The transport system realizes the transport of pallets between all other moduls.
- *high rack storage and rack feeder (C)*
  The high rack feeder is controlled by a Phoenix slot PLC, which also manages the content of the high rack storage.
- *manual workplace (D)*
  For the manual workplace there is of course no automation device needed, but a GUI showing the requested pattern and enabling the worker to communicate with the system.

For the realization of the demonstrator it was not only necessary to set up a complete PABADIS environment as in former states of testing, but to implement CMUs able to realize real control functionalities. This means to integrate real automation devices into the system. Furthermore some additional CMUs providing functionalities needed for the demonstrator such as a routing CMU and modifications at the Transport CMU had to be implemented. Within the demonstrators 3 different types (with respect to the automation device) of processing CMUs are used.

- JetterPLC JetControl (Com-linked CMU, based on a Jetter JetControl 241 PLC)
- Phoenix SlotPLC (Com/Os-linked CMU, PC with extension card)
- TimeSys real-time JVM (Java-linked CMU, RT-Linux PC)

Beside the processing CMU some logical CMUs or CMU-parts are needed to support the functionality of the demonstrators. For the realization of the transport within the demonstrators it is necessary to implement a routing CMU which is able to store all possible routes in the system with appropriate costs, calculate all possible routes between two points in the system and provide this information to the PA for planning its way through the CMU community. To enable an efficient testing it is very useful to have the possibility of a global overview of all running CMUs and PAs in a system. For this reason the observer CMU was developed. This CMU is updated continuously with all information from the LUS and agent-system and enables the user to monitor.

New requirements and experiences during the demonstrator design and tests lead to a lot of enhancements in several parts of the PABADIS-system. Although the most improvements resulted from specific problems and requirements, the proposed and implemented solutions are of general nature and increase the applicability of the PABADIS-concept in real industrial solutions:

- Blocking tasks and allocation filter mechanism for an improved scheduling on CMU-level.
- Synchronisation of PAs with respect to time and location conditions or combinations of both
- Extended MO/WO XML-data structure for the synchronization of tasks
- Modification of the FCM within the RAFI (new Transition to defer a task)

Objective 3.2 Verify this test against industry's requirements

The logic continuation of “soft” testing, “hard” testing and the evaluation of the developed software components was a verification of the results which was realised against the background of the end users and the IRG members’ experience.

The verification of the functionality and the compatibility of the PABADIS software components in conjunction with selected user partners was seen as a mapping of real industry scenarios to PABADIS’ facilities. In this context PABADIS solutions were given with specific PABADIS components, taking into account the costs that would potentially arise. For that reason cost models were introduced that were used for verification and assessment. Three cost models were developed; one cost model for bringing the different components into the market, one for an assessment of further research and one cost model for the implementation of a component into a running business.

In order to prove the applicability of the general PABADIS-concepts also in more complex scenarios than the demonstrators can provide, a simulation scenario was developed based on the Daimler Chrysler P2000+ plant in Stuttgart, Germany. The results of this simulation were compared with the simulation of a conventional system. By analysing 160 different simulation runs it was possible to state that a PABADIS system is superior to conventional systems with respect to production execution efficiency; derived from that it can also be stated that a PABADIS system provides also more flexibility than a conventional one.

By the given numbers is has been shown that a PABADIS system is able to react much faster and more efficiently on a turbulent environment and is able to ensure a more flexible production with decreasing production durations caused by waiting times.

The overall feedback from the verification and from the main target groups - the automation industry and the end user companies - is that the PABADIS approach and technology of a highly-flexible and adaptable automation architecture is very promising and valuable and further collaborations between the consortium on the one side and the automation industry and their end users on the other side should be strengthened.

For the exploitation and dissemination relevant goals of the project please refer to chapter 6 of the document at hand.

5 Deliverables and References

In the following the major deliverables, representing the main scientific and implementation work done within the project, will be briefly described.

Deliverable 1.1: Industrial Requirements and Overall Specification

This deliverable stands at the beginning of the PABADIS project. Its purpose was to give a common understanding and a common framework for the following tasks, which will be done by the definition of a common model. Also the industrial requirements concerning plant automation were collected and an update concerning the state of the art was done.

Within this document the fundamental research areas and industrial requirements for the solution of the PABADIS project are investigated. Based on a short description of the project aim, the state of the art and industrial requirements in the field, general solutions in industrial automation, automation hardware, communication and automation software are given. Based on the given statements, conclusions for the project are sketched and the architectural structures for the project are given. Furthermore, a naming of all relevant PABADIS components was done.

Deliverable 1.2: Agent Specification and Design

Deliverable 1.2 specifies and designs the agents of a PABADIS system, namely the Product Agents (PA), Residential Agents (RA), and Plant Management Agents (PMA), with respect to the technologies used for them. However, this kind of design and specification relates to the determination of an agent platform that fits into the PABADIS system requirements. For this reason, the current technologies have been evaluated, and based on industrial requirements choices for a suitable agent platform, appropriate agent design, and a communication scheme have been made.

The first step in the task was a performance evaluation and technological choice considering the agent platforms

- Voyager, Hive, Aglets, Grasshopper, Lana, Odyssey, Zeus, Agent Builder, Open Cybele, D'Agents, and Mozart

The structure, communication mechanisms and life cycle of PA, RA, and PMA are described and a short introduction into Agent Container concepts is given. Especially the Grasshopper agent technology and the Lana platform were investigated. The result of this investigation and comparison is the decision, that Grasshopper was used for the PABADIS project.

Deliverable 1.3 Platform Layer and Host Specification and Design

Within this deliverable several problems on designing and defining the platform layer are described. This contains a definition/description of the term "platform layer" and its functionality. Based upon this definition and the knowledge about its functionality the expected output of this design process has been fixed.

The platform layer provides a basis system on various hardware and software platforms of a distributed system. The programming language JAVA was chosen because the platform independency of Java makes it easily possible to provide programs, which follow the paradigm "Write once, run anywhere". Java programs do have the capability to be applicable on different platforms, required by industrial applications.

Furthermore the most important communication requirements in a (PABADIS) plant network are described and explained. This description lead to a communication scheme definition and, hence, to the definition of an appropriate communication protocol.

As the basis of a distributed system is the communication between its components, an appropriate communication scheme is defined, which fulfills the communication requirements of the participating components. Current and likely future communication requirements, including the PABADIS agent communication, have been analyzed. Based upon this, the communication scheme and, hence, the communication protocol was chosen. Furthermore, the Jini/Pini Services have been designed. This was done by describing the behavior of PABADIS entities as P&P services and clients. It was analyzed, which entity mainly acts as a service and which mainly acts as a client. Based on the given P&P components it was determined, how the clients use the services, how the CMUs provide their services and which kind of services are mandatory.

Deliverable 1.4 Agent Fabricator Specification and Design

This deliverable is one of the four documents that conclude Work Package 1 of the PABADIS project, dedicated to "Overall model design and specification". Objective of Task 1.4 was originally the "Conceptual design and specification of the Agent Fabricator". The specification of the Agent Fabricator was gradually refined during the task 1.4 progress, and the scope of its activities was progressively extended to cover all centralized functions at the execution level, and not only the generation of the PABADIS agents. For that reason, the term **Agent Fabricator** has been replaced by the term **Agency**, which is more representative for this extended functionality. The Agency is actually an entity that includes the Agent Fabricator.

This deliverable introduces the part played by the Agency in the overall PABADIS system as the main link between Enterprise Resource Planning (ERP) and shop-floor processes. A list of existing MES functionalities has been drawn and how these will be addressed and constituted with autonomous agents. Furthermore the functions expected from agents, the detailed process of agent generation and the interfaces between the different actors are described. Finally, a sketch of agent creation is given, based on a simple hypothetical case (demonstrator).

Deliverable 1.5 ERP Interface Specification and Design

This document as part of the specification and design of the overall model provides a general design for an appropriate interface for connecting the shop-floor with current ERP-technologies in order to exchange all needed information between these levels. The deliverable reflects the work carried out for the definition of the generic PABADIS-ERP-interface. This includes the investigation of the ERP-systems' potentials and standards and, vice versa, the identification of the requirements from an ERP system's side. Based on these recommendations the technical requirements for the development of the ERP-interface were given. Finally, the interface design is described with regard to existing technologies and interoperability aspects.

Deliverable 2.1 Development of the generic Agent V1

This deliverable serves the overall development of the MAS and therefore describes the structure of all three types of developed agents: PA, RA, and PMA in respect to the Agency and the CMUs. Based on this document the first version of a test bed (software demonstrator) is created. The implementation is based on the Grasshopper agent platform and involves the MAS, Agency, ERP, CMU-community and Look Up Service (LUS).

Deliverable 2.2 Development of the Generic Agent V2

This document provides an updated specification given in deliverable 2.1. It was written in order to describe the agent community in PABADIS and to use this document as a basis for the demonstrator development. The purpose of each type of agents, the Product Agent (PA) life cycle, the Residential Agent (RA) with methods and interfaces as well as the Plant Management Agent (PMA) were explained in detail. Furthermore - based on use cases developed to understand and verify the necessary communication mechanisms within the agent community and its interactions with other components of the system - the communication mechanisms within the agent community and interactions with other components of the PABADIS system are described here. Based on this document the final version of a test bed is created.

Deliverable 2.3 Development of the machine representation

The content of this deliverable is dedicated to describe the work done in and the reached results of task 2.3. With this task means were developed to present the capabilities of an individual machine to the residential agent. The representation will cover the machines' availability, capacity, capability, and other attributes. These attributes are represented in an abstract way. Parts of this information will also be offered via the PnP-system that announces the possibilities of each agent in the agent community.

Within this document an overview of the PA/PMA structure, the XML data (which are included in these agents) as well as the structure and behaviour of ERP and Agency are given, and the use of capability description, work order description, and information request to ensure the finding and use of the required functions by PA/PMA are described. In a short overview related standards within control and control related areas are mentioned. Based on that the amount of data required to specify the capabilities of a CMU, the work orders of PAs, and the information requests of PMAs are described and XML structures based on DTD as well as Schema are defined, able to cope with this data set. Finally, the requirements for the integration of an XML parser into the CMU structure are given. This parser is used to compare CMU capabilities and PA/PMA requirements.

Deliverable 3.1 Machine Platform implementation

The aim of this document was to describe the implementation of PABADIS on target systems. Because of the strong relationship to other deliverables this document is based on the results of del. 2.3, del. 3.3, and del. 6.3. This document spends most effort in the description of general reflections and generic concepts.

It includes a description of the common target system architectures, their relationships to PABADIS, and a proposal how to implement the concepts of del. 2.3 and del. 3.3 in general. Furthermore, this document describes a general approach how to implement the different CMU types.

In addition, this document includes a short description of the existing industrial automation systems that are envisioned to be used as a platform for PABADIS. At last, this document includes a guideline how to integrate these industrial automation systems into PABADIS.

Deliverable 3.2 Platform Testing

The aim of this deliverable is to describe the methods and concepts how to verify and validate the functionality and the performance of a PABADIS CMU.

Testing functions of MES, PA, PMA, LUS, Agency, or other PABADIS parts outside the CMU is out of the scope of this deliverable. This deliverable is one of the outcomes of Task 3.2. It spends most effort in the description of general reflections and generic concepts. The PABADIS specific parts are based on the results of del. 3.1, del. 3.3, del. 3.4, and del. 3.5.

Deliverable 3.3 Host Design and Development

This deliverable mainly describes the design of PABADIS hosts. This includes the overall description of the CMU components. The design goal for the host was a generic representation of the host platform to the agent community. Therefore, the most effort was spent in the definition of the generic representation of the CMU functions to the agent community. As a result fine grained definitions of the Host (CMU) components were given for:

- the RAFI (Residential Agent Function Interface) with different components: FCM, ECM, PI, CD and IRI
- the CFM (Common Feature Module) with the XML component and Schedule component.

Deliverable 3.4 Plug-and-Participate Capability

This document aims at providing a fine grained definition of the CFM's P&P module and includes:

- a definition of requirements
- the evaluation of security mechanisms
- the evaluation of different PnP technologies
- a detailed description of solutions for P&P use in PABADIS
- the introduction of an improvement of the Jini technology by a new matching algorithm, which is efficient and generic and fits best to the PABADIS requirements
- the description of the implementation of the PnP-module and first test with the agents

Deliverable 3.5 Communication Platform

This document describes the work done in Task 3.5 (communication platform). The main focus is on the mutual influence of PABADIS related messages with real-time automation messages. Therefore at first commonly used and newly developed automation communication concepts based on Ethernet have been assessed. Furthermore simulations of the network behaviour have been performed.

Based on the assessment and Task 3.2 (machine platform-integration and -testing) simulation results are described. One communication system was chosen, which was integrated together with the CMU software. This CMU was used to measure the communication behaviour.

Deliverable 4.1 Interface Development

This deliverable describes the work carried out in task 4.1. where a generic interface which is able to interconnect to as many ERP systems as possible is implemented, based on the design of Task 1.5. This interface is part of the Agency, therefore there was a close collaboration between this task and Task 4.2. as each task refers to the other. The deliverable provides a description of the needed interface for integrating the shop-floor of an industry with current ERP technologies in order to supervise and manage an industry. Deliverable 4.1 reflects the work carried out during task 4.1 concerning the development of a generic interface between ERP systems and the PABADIS infrastructure. During this task the main working activities and achievements that we describe also in the present document were:

- Investigation of Enterprise Application Integration tools and methods, which trends and functionality exist. Selection of the Functional Integration Model.
- Development of the Interface structure and functionality, as part of the Agency. An Interface that can integrate the PABADIS plant with the ERP systems is implemented. A description of the structure and functionality is given, but it must be seen in parallel with task 4.2, where the other components of the Agency are addressed.
- Specification of the Manufacturing Order Structure, XML common definitions, examples.
- Preliminary testing

Deliverable 4.2 Agent Fabricator V1

The implementation of the first version of the "Agent Fabricator" is the content of the deliverable 4.2. The initial vision of the Agent Fabricator has been slightly altered and refined all along the development of Work Package 1. As it was established in previous deliverables, the Agent Fabricator has been encapsulated within the Agency, an entity which is in charge of most centralised functions required or induced by the interface with ERP. This document deals with the following topics in detail:

- General description of data in manufacturing systems
- Agency design and implementation
- Agency data structures
- SCADA CMU implementation
- Simulation prototype

Deliverable 4.3 Agent Fabricator V2

This document describes the work done during the Implementation of the second and final version of the Agent Fabricator/ Agency. This task consequently completes and modifies the first version which was developed in Task 4.2. Since the initial vision of the Agent Fabricator/ Agency has been defined, some changes have brought up all along the development of Work Package 1. As it was established in previous deliverables, the Agent Fabricator has been incorporated in an Agency, an entity which is in charge of most centralised MES functions required or induced by the interface with ERP. The document describes the following aspects:

- ERP-Agency Communication
- Agency-Agents Communication
- ERP Interface Design
- Agency Design
- SCADA Interface Design

Deliverable 5.1 Soft Demonstrator Testing Report

The deliverable gives a report on the implementation process and progress of the test bed development including PA, RA, Agency, ERP, CMU, Function and LUS. The testing progress and testing results are given in this document, including coding, component testing and the system integration, where the communication between components are implemented and tested.

Starting with the overall system structure, including ERP, Agency, MAS, CMU and LUS the document covers the Agency design for the test bed as well as an overview of the Agent Community test bed and the results of test within these environments.

Deliverable 5.2/5.3 Hard Demonstrator Implementation Report

This document gives a report on the conception, implementation process and test results of the demonstrator development. This includes the demonstrator hard and software as well as additional components for a PABADIS-system, developed within the focus of work package 5. The main test results are given in this document, including single automation functions, additional components and the system integration, where the overall functionality of the demonstrators were implemented and tested. This document is written in order to describe the demonstrator structure and development process and gives an overview of the test results.

A detailed description of the overall requirements, hardware structure, modularisation as well as design of the PABADIS related parts - such as CMUs and Work Order for the Fischertechnik model demonstrator and for the experimental plant demonstrator - are given and newly developed components for the demonstrator implementation are explained.

Finally, the test results and PABDADIS–concept enhancements resulting from experiences and requirements of the demonstrators are mentioned.

Deliverable 5.4 Hard Testbed Verification

In order to achieve concurrent engineering, the testing and evaluation is done first in a “Soft” test bed and then in a “Hard” test bed. In order of this task to be more coherent, work has been carried out in parallel with task 6.2 for the assessment and evaluation of costs for potential users of the PABADIS components. This deliverable is a common presentation of the results achieved during tasks 5.4 and 6.2 for verifying the results of the “soft“ and “hard” testing. The evaluation of the developed software components was verified against the background of the end users’ and the IRG members’ experience, considering costs as a key parameter of the verification. A cost model is described for bringing the different components into the market or for the evaluation of further research activities and another cost model for the implementation of a component into a running business. Furthermore, there is a description of the evaluation of component costs for issuing them to the market or proceed to further research and possible scenarios of usage. This description is given for each of the developed PABADIS components.

Finally, the team accomplished a verification of the functionality and the compatibility of the PABADIS software components in co-operation with selected user partners. This verification is seen as scenario practises to industries under consideration of different problems; here PABADIS solutions are given with regard to a specific component, taking into account the potential cost. These cases contain a description of the adaptation of the system, advantages of the PABADIS system against the old system and an evaluation of system re-engineering costs to reach the PABADIS system.

Deliverable 5.5 Soft Testbed Complex System Simulation

Following a recommendation of the 2003 review, a PABADIS computer simulation was designed and realised to improve the possibility to test more complex scenarios of a PABADIS-plant than it was possible by means of the rather limited size of the hard-demonstrators (compared to a whole plant).

Against this background, this deliverable describes design, set up, and evaluation of a corresponding simulation which was done additionally to the two existing demonstrators.

Deliverable 9.1 Relevance of PABADIS for the Fast Moving Consumer Goods Industry

Within this first deliverable of the Swiss members of the PABADIS consortium the relevance of the PABADIS concept for the production and packaging equipment industry for the fast moving consumer goods industry was assessed and examined. The assessment of the relevance for the production and packaging equipment industry is eminent, as Italy, Germany and Switzerland generate together more than 36 billion EUR sales in this industry and employ around 115’000 skilled engineers and technicians. The first part of this document outlines the packaging equipment industry. Facts such as the high diversification and specialization are described. It is also shown that the FMCG industry seeks standardization and high throughput and that within the industry, flexibility is circumstantial.

The second part compares PABADIS with the PackML initiative. PackML is the packaging equipment manufacturers’ answer to the FMCG industry requirement to provide plug-and-play concepts for their production and packaging equipment. PackML focuses on horizontal connectivity and seeks to define standards for vendor independent inter-equipment communication.

Deliverable 9.2 Behaviour Modelling in PABADIS – A Case Study

This deliverable is written in correspondence to the PABADIS work plan of WP9. Its content is dedicated to the task “Behaviour Modelling in PABADIS”, which aims to extend the application areas of PABADIS by introducing agent behaviour modelling.

The document starts with an introduction, which reveals extension capabilities of PABADIS. In order to identify the behaviour, application areas and limitations of PABADIS, different types of manufacturing systems, manufacturing environments and product types are classified. The classification and the observations are the base for the discussion on industrial application fields, further research issues and recommendations concerning behaviour modelling. Then the model based software development method CIP (Communication Interacting Processes) is introduced. This method has been chosen in order to provide PABADIS agents and other components of PABADIS with explicit behaviour. The method was described in general, followed by an overview of the most important elements of CIP. Furthermore, the PABADIS behavioural extension will be explained. Finally a case study was presented addressing the problem domain of applying PABADIS to unbuffered serial production lines with multiple routes.

The conclusion of this task is, that extending generic agents and other components of PABADIS with implementation independent behaviour models is a valid contribution to PABADIS. The proposed approach allows to solve difficult manufacturing problems, with reduced risk of fundamental design errors and simplified maintenance.

Deliverable 9.3 Development of a Java based CMU

This deliverable is the 3rd within the scope of Work Package 9 of the PABADIS project. It is dedicated to the development of a Java-based CMU, in order to gain experience with Java on shop floor. The main focus of this task is on the level of machine control and therefore in wider sense on mechatronic systems.

The document starts with a general description of the robot which was used as test bed for the case study at hand. Therefore the construction, the mechanical specification, the electrical control cabinet and the PC-based control system will be explained. Further, this document includes the software development approach as subsequent listed:

- SafetyManager FSM: Machines on shop floor ought to be safe regarding software, as potential failure could cause human injury or machine damage. As safety was considered to be one of the main requirements for this case study, a safety manager has been developed.
- Kinematics: Robot kinematics in general describes the transformation from joint space to Cartesian coordinate space in both directions. Within this chapter the mathematical approach for the implementation of the kinematics module for the robot will be described.
- Axes Controller: This chapter describes the control engineering approach which builds the base for the implementation of the realtime axes control of the robot.

At last, the implementation is documented. The developed software can be clustered in a generic and a specific part. The generic part consists of general purpose Java classes for robotics and a native library, which supports I/O and RTOS specific function access. The specific part can only be used for the case study robot. The last chapter gives an overview of the applied framework and describes the application level implementation, based on the considerations of the preceding chapters.

6 Project Management and Co-ordination Aspects

6.1 Overview

Project co-ordination was done in a very close relation of the regional project co-ordinators with the corresponding IMS project's Interregional Co-ordinator (ICP) Eckehardt Klemm[1], This close partnership between industry and academia was a surety for innovation and great technical advancement on the one hand and industrial usability on the other hand.

Each regional project had its own project management. Each regional project co-ordinator contributed to the management tasks. However, the project co-ordinators of the European project Dr.-Ing. Kai Lorentz[2] (until 2001) and Dr.-Ing. Axel Klostermeyer[2] took over a significant part of the IMS project management and contributed to the project establishment and operation.

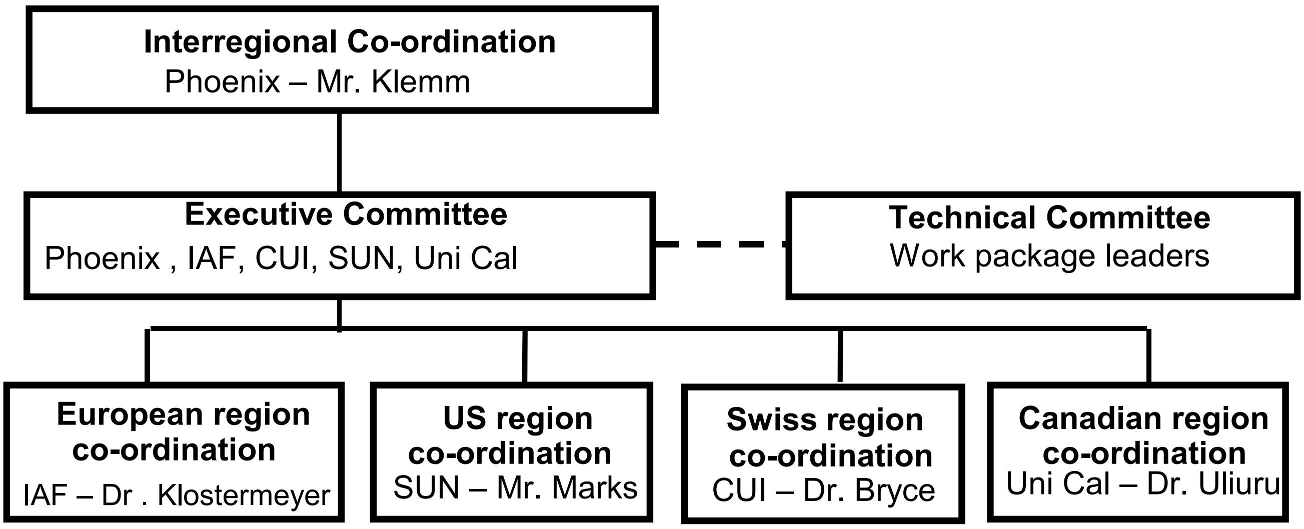

Figure 13: IMS project management structure

6.2 Performance and Dedication of the Consortium

First of all project management would like to state that the performance of the consortium and the dedication to the project were outstandingly well and the final results were better than expected. As the project's objective was more a successful "prove of concept" of a totally new architecture than the development of new products, there were no disagreements or even conflicts between partners which might have resulted from different views of the project results' exploitation and corresponding legal questions. What could be seen after a short time were (cultural) differences between the single partners and the way they contributed to the project's progress. Nevertheless, after a short time the consortium got accustomed to this and adapted itself to different working styles. So that, last but not least, the project did not suffer from this. This was especially due to the partners of work package 3 (University of Marburg, Phoenix Electronics, Jetter, University of Magdeburg) who in the course of the project became a driving force for the research, development and implementation work and in this function also significantly influenced the work of the other partners. Especially these above mentioned partners showed a very substantial dedication to the project which was more than "only" fulfilling a contract. Probably this can be ascribed to the different possibilities of the partners to exploit the project's results after its termination.

[1] From Phoenix Electronics, Bad Pyrmont, Germany

[2] From the Center Distributed Systems (CVS) of the Institute for Ergonomics, Manufacturing Systems and Automation of the Otto-von-Guericke University Magdeburg, Germany,

Language barriers in general could be neglected as all partners could agree on one common language (English) which was spoken very well by (nearly) each person working in PABADIS. Neither was the geographical distance between the single partners an obstacle for the project's progress as modern communication media (telephone and e-mail) could be used and the PABADIS travel budget enabled periodical physical meetings as well of the whole consortium as of single working groups.

6.3 Encountered Challenges

The project, generally speaking, did not encounter serious challenges or even problems; the consortium was well composed and worked together as it realistically could be expected. Slight deviations from the original project plan had to be faced when the demonstrators were implemented: When comparing planned activities and actual work in period 4 (01/06/2002-30/11/2002) it had to be stated that - on the one hand - the work which was intended to do was dealt with and even more than that was done, as several special issues were handled additionally – e.g. concepts for transportation, tooling, robustness. On the other hand, the formerly intended time schedule was left and a prolongation of the project of approximately ½ year was asked for and granted. The detailed reasons for this were the following ones:

- There was a necessity to develop a derivative of the originally intended Jini-technology which is suitable also for small devices;
- It turned out to be reasonable to design an abstract layer between the PABADIS system-entities, e.g. a CMU, and the used agent system to guarantee the generic character of the PABADIS architecture;
- The consortium needed more iterations for testing and adaptation then originally intended due to an enlarged function set as a tooling strategy and a robustness concept;
- The extension offered the chance of a longer parallel work between the international IMS-PABADIS consortia from Switzerland, Canada, USA, and Europe.

Nevertheless, this delay was never seen as critical to the project by the PABADIS management.

What furthermore turned out to be less effective to PABADIS' progress were the non-synchronised start dates of the respective projects in the other IMS regions Canada, Switzerland, and USA. But as this is an IMS topic, it shall not be dealt further with at this point.

6.4 Co-operation with Industry, other Projects, and Dissemination

6.4.1 The PABADIS Industrial Reference Group

To assure the practical applicability of the PABADIS project a strong collaboration with industry was searched for. For this reason an Industrial Reference Group (IRG) was established. The Industrial Reference Group (IRG) served the purpose of broadening the project's scope and experience without increasing the technical and management risk and work load with too many different test cases within one project. The work of the IRG was mainly performed during workshops, attended by the IRG members and the project participants. During these workshops, the project status and issues were discussed according to the project's progress. Especially in France, Germany, and in Italy the consortium hereby concentrated on major companies such as SIEMENS, Schneider, KUKA, FIAT and SAP as well as on SME's such as WEROS GmbH. In the field of the academic research contacts were established resp. strengthed with FH Harz and IWU Chemnitz.

Summarising it can be stated that the outcomes of these visits were generally positive and very useful for the project work.

6.4.2 Consideration of related IMS projects

The following IMS projects were looked deeper into or contacted for an exchange of experience:

Project Title, Acronym	Project topic
Holonic Manufacturing Systems, HMS	Development of a multi-agent structure for control based on residential agents

Table 3: Related Projects

Especially to parts of the HMS consortium (Schneider Electric, University of Calgary) deeper contacts were established as this project seemed to be of deeper importance for PABADIS as well from the side of the partners involved as from side of their contents. By the involvement of former Canadian HMS partners into the Canadian IMS PABADIS project even a formal basis for this collaboration could be established.

6.4.3 Consideration of related EU Projects

The following EU projects were looked deeper into or contacted for an exchange of experience:

Project Title, Acronym	Project topic
Advanced Infrastructure for Pan-European Collaborative Engineering, E-COLLEG	Use of enabling technologies (JINI, Corba, RMI) to generate generic collaborative services
Mobile Agent environments in Intelligent Manufacturing, MARINE	Investigation of enhancements to the IN architecture by means of flexible allocation of resources
Communication Agents for Mobility Enhancements in a Logical Environment of Open Networks, CAMELEON	Enhancement of service and management architectures in communication systems by mobile agent systems
An agent based approach to controlling resources in UMTS networks, SHUFFLE	Novel architecture for efficient, scalable and robust real time control of resources in mobile systems
Federate European Tourism Information System Harmonization - Engineering Task Force, FETISH - ETF	Development of a Jini based spontaneous networking structure to enable a tourism information system, based on servers and clients without a direct network structure.
Distributed Advanced Fetish Network Evolution, DAFNE	The overall goal of DAFNE is to enhance the effects of the Fetish network results, bringing the project to an industry grade that can be exploited on the market through business and marketing actions, aimed at building up a real commercial activity.
Lightweight Extensible Agent Platform, LEAP	Project LEAP developed an agent platform that is: lightweight, executable on small devices such as PDAs and phones; extensible, in size and functionality; operating system agnostic; mobile team management application enabling, supporting wireless communications and TCP/IP; FIPA compliant.

Table 4: Related Projects

Especially to parts of the LEAP project (Siemens CT) deeper contacts were established as this project seemed to be of deeper importance for PABADIS as well from the side of the partners involved as from side of their contents.

6.5 Technology transfer and dissemination

Technology transfer between the different regional projects has taken place at various levels. The Swiss project was fully integrated in the European project and therefore the European and Swiss partner have equal access to the common project results. This was technologically realised by a common data space for deliverables with unique access rights for all project partners. Exchange with the US and the Canadian project took place on demand. Additionally in March 2003 a three day technology workshop was held Vienna where the experts exchanged technology information in detail.

Dissemination of the PABADIS results has been carried out with journal papers, book contributions, presentations on conferences and exhibitions. The international PABADIS project was presented in a special workshop at the IEEE INDIN03 conference in Banff and on different further conferences by individual papers.

7 Outlook

The PABADIS project has been a successful prove of concept for agent based order control systems within factory automation, replacing conventional MES systems. Main parts and results have been (as depicted above) a generic interface structure to ERP level systems, a generic, agent based methodology for order evaluation, resource selection and resource allocation, as well as a generic access path to conventional control devices controlling the allocated resources for order processing. Minor results are experiences in the design and application of multi agent systems, new architectures for field control system design, new Java based control devices, and XML based generic process and product descriptions.

The PABADIS project partners intend to exploit the mentioned results within further activities in different ways. Since the project was a proof of concept the main activities will be within further research and development ranging from theoretical, academic research to development of structures and architectures. A direct exploitation of the project results within product design will play a minor role, at least short termed.

Some partners of the PABADIS consortium intend to extend the project activities within a new IMS project improving the PABADIS results with respect to distributed intelligence and new upcoming technologies for data storage and transmission. Therefore, also new partners will be integrated in the project activities.

8 Conclusions

Objective of PABADIS was to enable a plug-and-participate-environment with highest adaptability and flexibility from the ERP -System to the single machine control in single piece production plants; mobile agent technology here played a major role. In this way the dynamics which in modern management and organisation concepts are standard meanwhile, were realised on the technical/ informational level as well.

To reach this objective an international consortium with companies and universities from Canada, Europe, Switzerland, and USA was established under the umbrella of the IMS (Intelligent Manufacturing Systems) programme. The European PABADIS consortium consisted of companies and universities from Austria, France, Germany, and Greece – lead by the Center Distributed Systems at the Institute of Ergonomics, Manufacturing Systems, and Automation of the Otto-von-Guericke-Universität Magdeburg, Germany. Though (a) the different IMS regions had different project start dates which complicated a synergetic co-operation and (b) the different cultures and interests of the various partners also sometimes raised small conflicts between them, the consortium finally was able to beat the original expectations and was able to deliver more than what was promised within the 1999 project plan.

Note: IMS helped us to build an interregional PABADIS R&D network, and I consider this a strong point of IMS, but we had unnecessary delay to get the regional projects running under the IMS framework. This might have been due to the fact, that we did not know and care enough about the regional specialities at the beginning. After having learnt the difference of the IMS regions and especially the different funding and ratification procedure within the IMS regions we managed that problem. But, somehow this different regional funding and ratification procedures seem to be a weak point of IMS which implies the tendency of asynchronous and self contained regional project accomplishment and eventually loosing sight of the international collaboration.

As the initially mentioned was a very ambiguous objective which is reaching far beyond the current state of the art, “only” a prove of concept was originally planned and no final products were aimed at. Nevertheless, as the demonstrators are running very well, the consortium would be able to implement this concept professionally in an appropriate industrial environment. The challenge of selling such a new system the first time without any references running is that the PABADIS system relies upon IT technologies as Java based agent technology, Ethernet, and PC based controls. These technologies so far are not standard in production environments and a certain caution can be observed at machine builders and plant erectors towards this “paradigm change”. While the use of PC based field controls, e.g. for robots, is not exceptional anymore, the use of Java and the employment of Ethernet TCP/IP as universal bus system are just in their infancy. Against this background one of the big end users from the automotive area told the consortium: ”Such new automation architectures are requested. But the risk in being the first one to implement it is too high.”

To overcome (a) this market entrance barrier as much as possible and (b) to design the concept as much as possible according to industrial needs, an Industrial Reference Group (IRG) was establish in parallel to the technical research and development activities. The work of the IRG was mainly performed during workshops, attended by the IRG members and the project participants. During these workshops, the project status and issues were discussed according to the project's progress. Especially in France, Germany, and in Italy the consortium hereby concentrated on major companies such as SIEMENS, Schneider, KUKA, FIAT and SAP as well as on SME’s such as WEROS GmbH. In the field of the academic research contacts were established resp. strengthed with FH Harz and IWU Chemnitz.

As final conclusion of this report - and of the PABADIS project as such - its internationtional coordinating partner and the regional project co-ordinators would like to state that though a lot of work has still to be done to transfer this “prove of concept” research project into a commonly used product. This barrier should not block the view on the huge potentials this “Web Integrated Manufacturing” offers to the European end-users of automation technology.

9 Annex

9.1 Dissemination

9.1.1 Journal papers and book contributions

A. Lüder, J. Peschke, T. Sauter, S. Deter, D. Diep, “Distributed intelligence for plant automation based on multi-agent systems – the PABADIS approach”, *Journal on Production Planning and Control*, in print.

E. Klemm, A. Lüder, “Agentenbasierte Flexibilisierung der Produktion bei Verwendung von vorhandenen Steuerungssystemen”, *atp (Automatisierungstechnische Praxis)*, in print.

P. Massotte, J. Reaidy, Y. Liu, D. Diep, “Intelligent Agents for Production Systems”, *Journal Européen des Systèmes Automatisés (JESA)*, In print.

K. Lorentz, A. Klostermeyer, “PABADIS Project envisions plants with decentralized intelligence”, *Advanced Manufacturing Technology Alert* (2000), John Wiley & Sons, pp. 2-3.

H. Kühnle, K. Lorentz, A. Klostermeyer, “Neue Wege in der Fabrikautomatisierung”, *Werkstattstechnik* 91 (2001), pp. 138-141.

A. Klostermeyer, K. Lorentz, “WIM - Web Integrated Manufacturing”, *Praxis Profiline: Industrial Ethernet*, Vogel Verlag, Würzburg, April 2001, pp. 14-18.

K. Lorentz, A. Klostermeyer, “GNOSIS VF – An Account And Outlook” in B. Stanford-Smith, E. Chiozza, M. Edin (eds.), *Challenges and Achievements in E-Business and E-Work*, part 2, IOS press, Amsterdam, 2002, pp. 1344 – 1349.

C. Schwab, J. Peschke, K. Lorentz, “Echtzeit-Java in der Steuerungstechnik- Implementierung verschiedener Übertragungsprotokolle und deren Anwendung in der Steuerungstechnik”, *A&D Kompendium 2003*, publish-industry Verlag, Jan. 2003, pp. 44 – 45.

A. Klostermeyer, E. Klemm, Multi Agent based Architecture for Plant Automation, in: R. Zurawski [Ed.]: Industrial Information Technology Handbook, CRC Press, 2004

J. Peschke, A. Lüder, Java and Industrial Applications, in: R. Zurawski [Ed.]: Industrial Information Technology Handbook, CRC Press, 2004

9.1.2 Conference papers

K. Lorentz, A. Klostermeyer, „Paradigmenwechsel in der Fabrikautomation“, in K. Bender, G. Brandenburg, R.D. Schraft (eds), *Proceedings of SPS/IPC/Drives 2000*, Nürnberg, 28.-30. Nov. 2000, Hüthig, pp. 41 – 48.

H. Kühnle, A. Klostermeyer, K. Lorentz, “A Paradigm Shift to Distributed Systems in Plant Automation”, *International NAISO Congress on Information Science Innovations (ISI' 2001)*, Dubai, 17.-21. March 2001, pp. 463-469.

S. Deter, K. Sohr, “Pini – Plug&Play for KVM/CLDC”, in Th. Böhme, H, Unger (eds), *Innovative Internet Computing Systems, International Workshop IICS 2001*, Ilmenau, Germany, 21.-22. June 2001, Springer, pp. 53 – 66.

P. Massotte, P. Couturier, Y. Liu, J. Reaidy, “Business network: rethinking supply chain management systems of virtual factories”, *International Conference on Industrial Engineering and Production Management (IEPM 2001)*, Québec, Canada, 20.-23. Aug. 2001, pp. 965 – 971.

S. Deter, “Plug&Participate für ‘limited devices’ mittels Gateway-Strukturen”, *Net.ObjectDays 2001*, Erfurt, Germany, 10.-13. Sep. 2001, pp. 130 – 137.

T.Sauter, P. Palensky, “Network Technology for Distributed Plant Automation”, *IEEE International Conference on Intelligent Engineering Systems (INES)*, Helsinki - Stockholm, 16.-18. Sep. 2001, pp. 407-412.

J. Reaidy, P. Massotte, Y. Liu, D. Diep, “Product and process reconfiguration based on intelligent agents”, *Technology, Economic and Social Applications of Distributed Intelligence Workshop (TESADI) – IEEE International Conference on Systems, Man, and Cybernetics (SMC’01)*, Tucson, AZ, 7.-10. Oct. 2001, pp. 3397 - 3402.

S. Deter, “JPGateway – Plug&Participate für ‘limited devices’”, *APC 2001*, Munich, Germany, 10.-12. Oct. 2001, pp. 35 – 44.

T. Sauter, P. Massotte, “Enhancement of Distributed Production Systems through Software Agents”, *IEEE International Conference on Emerging Technologies and Factory Automation (ETFA)*, Antibes Juan-les-Pins, France, 15.-18. Oct. 2001, pp. 267-272.

S. Deter, “Plug&Participate for Limited Devices in the Field of Plant Automation”, *IEEE International Conference on Emerging Technologies and Factory Automation (ETFA)*, Antibes Juan-les-Pins, France, 15.-18. Oct. 2001, pp. 273-278.

P. Massotte, Y. Liu, J. Reaidy, “DAPS : Dynamic Adaptation of Complex Production Systems”, *13th European Simulation Symposium in Simulation Industry (ESS’01)*, Marseille, France, 18.–20. Oct. 2001, pp. 965 - 972.

P. Massotte, “Virtual factory and business to business. Network : rethinking Manufacturing execution systems & Supply chain execution”, 5ème Conférence Internationale de Management des Réseaux d’Entreprises (CIMRE), Mahdia, Tunisia, 25.-26. Oct. 2001.

E. Klemm, A. Lüder, „PABADIS – An Infrastructure for flexible Shop Floor Automation“, *4th International IFAC Conference on Fieldbus Systems and their Applications (FeT 2001)*, Nancy, France, 15.-16. Nov. 2001, pp. 121 – 124.

A. Klostermeyer, J. Peschke, „Flexible Auftragsnavigation mit mobilen Softwareagenten“, in R.D. Schraft, K. Bender, G. Brandenburg (eds), *Proceedings of SPS/IPC/Drives 2001*, Nürnberg, 27.-29. Nov. 2001, Hüthig, pp. 218 – 228.

K. Lorentz, J. Peschke, A. Lüder, „Echtzeit-Java in der Automatisierungstechnik – Spezifikationen und praktische Anwendung“, in R.D. Schraft, K. Bender, G. Brandenburg (eds), *Proceedings of SPS/IPC/Drives 2001*, Nürnberg, 27.-29. Nov. 2001, Hüthig, pp. 440 – 447.

K. Lorentz, A. Lüder,J. Peschke, „PABADIS – Fabrikautomatisierung basierend auf verteilten Systemen“, in R.D. Schraft, K. Bender, G. Brandenburg (eds), *Proceedings of SPS/IPC/Drives 2001*, Nürnberg, 27.-29. Nov. 2001, Hüthig, pp. 243 – 251.

D. Diep, P. Massotte, J. Reaidy, Y. Liu, “Design and Integration of Intelligent Agents to Implement Sustainable Production Systems”, *2nd International Symposium on Environmentally Conscious Design and Inverse manufacturing (ECODESIGN 2001)*, Tokyo, Japan, 11.-15. Dec. 2001, pp. 729 - 734.

D. Diep, P. Massotte, R. Bataille, “Intégration d‘agents intelligents dans un système de production”, *NIMESTIC 2001*, Nimes, France, 12.-14. Dec. 2001, pp. 53 - 57.

Y.K. Penya, T. Sauter, “Network Load Imposed by Software Agents in Distributed Plant Automation”, *IEEE International Conference on Intelligent Engineering Systems (INES)*, Opatija, 26.-28. May 2002, pp. 339-343.

A. Lüder, J. Peschke, S. Deter, A. Bratoukhine, “Interaction of Software Agents and Real-Time Industrial Control System within a PABADIS System”, *14th Euromicro International Conference on Real-Time Systems*, Vienna, Austria, June 19.-21. June 2002, Work-in-Progress Proceedings, pp. 21-25.

S. Deter, B. Krzensk, T. Sauter, “Plug-and-Play for Semantic Fieldbus Networks”, *IEEE International Workshop on Factory Communication Systems (WFCS)*, Västerås, 28.-30. Aug. 2002, Work-in-Progress Proceedings, MRTC Report No. 61, August 2002.

Y.K. Penya, S. Mahlknecht and P. Roessler "Lightweight Agent Platform for High-Performance Fieldbus Nodes". In the Work-in-Progress Proceedings of the 4th IEEE International Workshop on Factory Communication Systems, WFCS 2002, Aug. 2002, Västerås Sweden, pp. 9-13

J. Reaidy, P. Massotte, D. Diep, "Negotiation and Decision-Making in Agent-Based Manufacturing Systems", *25th German Conference on Artificial Intelligence (KI2002)*, Workshop on Cognitive Agents, Aachen, Germany, 16.-20. Sep. 2002, pp. 73 - 87.

A. Bratoukhine, Y.K. Penya, T. Sauter, "Intelligent Software Agents in Plant Automation", *NAISO Symposium on Engineering of Intelligent Systems (EIS)*, Malaga, 24.-27. Sep. 2002, pp. 77.

Y.K. Penya, A. Bratoukhine, T. Sauter, "ERP Integration into Generic Plant Automation Model", *Workshop on Information Systems for Mass Customization (ISMC'2002)*, Malaga, 24.-27. Sep. 2002, pp. 137.

A. Klostermeyer, A. Lüder, "A Re-configurable Multi-Agent Based Architecture to Enhance the Flexibility of Job control in Single Piece Production Systems in Turbulent Environments", in C. Brom, E. Streiff, *Msy'02 Embedded Systems in Mechatronics – Proceedings of the 2nd Mechatronic Systems International Conference*, Winterthur, Switzerland, 3.-4. Oct. 2002, pp. 131 – 138.

J. Peschke, "Flexibilität und Durchgängigkeit in derFabrikautomation durch Java auf der Steuerungsebene", in C. Brom, E. Streiff, *Msy'02 Embedded Systems in Mechatronics – Proceedings of the 2nd Mechatronic Systems International Conference*, Winterthur, Switzerland, 3.-4. Oct. 2002, pp. 77 – 83.

D. Tektonidis, A. Vontas, U.Hess, J. Meschonat, "Handling the Shop-Floor of an Industry through ERP systems – A functional Integration Model", *e.2002: e-business and e-work*, Prague, Czech Republic, 16.-18. Oct. 2002, IOS press, Part 2, pp. 1350 – 1356.

R. Blume, S. Deter, "Generic Machine Representation in the PABADIS Community", *e.2002: e-business and e-work*, Prague, Czech Republic, 16.-18. Oct. 2002, pp. 1134 – 1140.

E. Klemm, "Generische Maschinenrepräsentation in der Welt von PABADIS", in K. Bender, G. Brandenburg, R.D. Schraft (eds), *Proceedings of SPS/IPC/Drives 2002*, Nürnberg, 26.-28. Nov. 2002, Hüthig, pp. 219 – 227.

S. Deter, A. Lüder, J. Peschke, "XML und Plug-and-Participate zur Flexibilitätserhöhung in der Fabrikautomatisierung", in K. Bender, G. Brandenburg, R.D. Schraft (eds), *Proceedings of SPS/IPC/Drives 2002*, Nürnberg, 26.-28. Nov. 2002, Hüthig, pp. 275-283.

Y.K. Penya, S. Detter, T. Sauter "Plug-and-Participate Functionality for Agent-Enabled Flexible Automation", IEEE International Conference on Intelligent Engineering Systems,INES'03, Mar. 2003, Assiut Egypt, pg. 136

E. Klemm, A. Lüder: Agentenbasierte Flexibilisierung der Produktion bei Verwendung von vorhandenen Steuerungssystemen, Automatisierungstechnische Praxis - ATP, Oldenburg Verlag, 4/2003, S.66-75.

Y.K. Penya, T. Sauter, Communication Issues in Multi-Agent-Based Plant Automation Scheduling. In Intelligent Systems at the Service of the Mankind, p. 135-144, Volume 1. (W. Elmenreich, J. A. T. Machado, I. K. Rudas, Eds.), Ubooks, Regensburg, Germany, 2003

A. Klostermeyer, E. Klemm, An Agent Based Flexible Manufacturing Concept, IEEE International Conference on Industrial Informatics INDIN'03, Banff 21.-24. Aug. 2003

J. Peschke, Real-Time-Java for Industrial Controls in Flexible Manufacturing Concepts, IEEE International Conference on Industrial Informatics INDIN'03, Banff 21.-24. Aug. 2003

M. Drouineaud, A. Lüder, K. Sohr: A Role based Access Control Model for Agent Based Control Systems, IEEE International Conference on Industrial Informatics INDIN'03, Banff 21.-24. Aug. 2003

A. Bratoukhine, T. Sauter, J. Peschke, A. Lueder, A. Klostermeyer, Distributed Automation – PABADIS vs. HMS, IEEE International Conference on Industrial Informatics INDIN'03, Banff 21.-24. Aug. 2003

D. Diep, P. Massotte, A. Meimouni, A. Bratoukhine, A Distributed Manufacturing Execution System Implemented with Agents: the PABADIS Model, IEEE International Conference on Industrial Informatics INDIN'03, Banff 21.-24. Aug. 2003

A. Klostermeyer, A. Lüder, Flexibilisierung der Auftragssteuerung auch bei kleinsten Losgrößen, A&D Kompendium 2004, S.38 ff.

A.Lüder, J. Peschke, A. Klostermeyer , A. Bratoukhine, T. Sauter: Distributed Automation: PABADIS vs. HMS, submitted to IEEE Transactions on Industrial Electronics, 2004.

9.1.3 Presentations without conference proceedings, exhibitions

IMS, ALTEC, "Integration Platform for Heterogeneous Information Systems", CeBIT 2002, EU Company Contact Point, Hannover, Germany, 17.-19. March 2001.

IAF, Hanover Industry Fair 2001, Hanover, April 23rd - 28th 2001

IAF, Workshop at Fraunhofer IFF, September 2001

IMS, World PC Expo, Tokyo, September 2001.

IAF, SPS/IPC/Drives 2001, Nürnberg, November 27th - 29th 2001

U. Hess, A. Vontas, "Implementing the plug-and-participate concept through ERP systems", *International Conference Software & Systems Engineering and their Applications (ICSSEA 2001)*, Paris, France, 4.-6. Dec. 2001.

IAF, Hanover Industry Fair 2002, Hanover, April 15th - 20th 2002

IAF, Canadian Gain/ FIPA Meeting, May 6th -10th 2002

IAF, SPS/IPC/Drives 2002, Nürnberg, November 26th - 28th 2002

IAF, Hanover Industry Fair 2003, Hanover, April 7th – 12th 2003

P2I, Industry and Trade Institute, Nîmes, October 3rd 2003

9.1.4 Conference Work Shop Organisation

E. Klemm, A. Klostermeyer, A. Lüder Organisation of the Workshop „Technologies for Flexible Manufacturing (TFM)" on 1st IEEE International Conference on Industrial Informatics INDIN'03, Banff 21.-24. Aug. 2003

9.2 IMS Project Meetings

Date	Place	Subject	Participants
00/04/22+23	Burlington, USA	IMS project setup	SUN, IAF
00/11/6+7	Magdeburg, Germany	European Project Kick Off	IAF, IMS, Unisoft, ICT, Jetter, Hatzopoulos, PUMI-I, Phoenix, P2I, EMA, SUN, (Univ. of. Osaka, Univ. of Tokyo)
01/02/01+02	Burlington, USA	US PABADIS project	Phoenix, SUN
01/05/22	Murten, Swizerland	Swiss PABADIS project	IAF, Pebble-Age, CUI University of Geneva
01/06/06+07	San Francisco , USA	US PABADIS project	SUN, Ajile, IAF, PUM-I
02/02/20	Braunschweig, Germany	Preparation Canadian PABADIS project	IAF, Phoenix, University of Calgary
02/04/13	Lake Louise, Canada	Canadian PABADIS project setup	IAF, Phoenix, PsiNaptic, University of Calgary
02/10/11	Erfurt, Germany	Coordination between EU und Canadian project	Phoenix, University of Calgary
03/01/27	Winterthur, Switzerland	Project meeting	Altec, EMA, Hatzopoulos, IAF, ICT, IMS, Jetter, Pebble-Age, PUM-I, Phoenix, Unige-CUI, ZHW
03/04/03_05	Vienna, Austria	PABADIS technology workshop	EMA, IAF, ICT, IMS, Jetter, Pebble-Age, Phoenix, Unige-CUI, ZHW
03/08/	Banff, Canada	Project meeting, PABADIS workshop at IEEE INDIN03 conference	EMA, IAF, Phoenix, University of Calgary, ZHW