

## A proposition of a manufactronic network approach for intelligent and flexible manufacturing systems

Fernando Almeida<sup>a\*</sup>, Paulo Dias<sup>a</sup>, Gil Gonçalves<sup>a</sup>, Michael Peschl<sup>b</sup> and Michael Hoffmeister<sup>c</sup>

<sup>a</sup>Faculty of Engineering of University of Porto, DEEC/DEI, Rua Dr. Roberto Frias s/n, Porto, Portugal

<sup>b</sup>Harms & Wende GmbH, Grossmoorkehre 9, Hamburg, Germany

<sup>c</sup>Manufacturing Engineering and Automation (IPA), Fraunhofer Institute, Stuttgart, Germany

### ARTICLE INFO

#### Article history:

Received 19 May 2011  
Received in revised form  
May, 31, 2011  
Accepted 31 May 2011  
Available online  
31 May 2011

#### Keywords:

*Intelligent manufacturing*  
*Production unit*  
*Quality systems*  
*Workflow models*

### ABSTRACT

The XPRESS project introduces a completely new scalable concept of a manufactronic networked factory, which is composed by a co-ordinated team of specialized autonomous objects (Manufactrons), each knowing how to do a certain process optimally. This knowledge based concept integrated the complete chain: production configuration (decrease of ramp-up time of at least 50%), multi-variant production line (varying types and volumes on a single line) and 100% quality monitoring. The manufactronic networked architecture allows continuous process improvement, and will be able to anticipate and to respond to rapidly changing consumer needs, producing high-quality products in adequate quantities while reducing costs. This concept is demonstrated in the automotive, aeronautics and electrical industry but can be transferred to nearly all production processes.

© 2011 Growing Science Ltd. All rights reserved

## 1. Introduction

### 1.1. In The concept of intelligent manufacturing systems

Global competition and rapidly changing customer requirements are forcing major changes in the production styles and configuration of manufacturing organizations. Increasingly, traditional centralized and sequential manufacturing process planning, scheduling, and control mechanisms are being found insufficiently flexible to respond to changing production styles and high-mix low-volume production environments (Shen, 1999). The traditional approaches limit the expandability and reconfigurability of the manufacturing systems (Sanchez, 2001). The centralized hierarchical organization may also result in much of the system being shut down by a single point of failure, as well as plan fragility and increased response overheads (Yang, 2003).

In the last twenty years, manufacturing concepts have had several redefinitions. In the eighties, the concept of flexible manufacturing systems (FMC) was introduced to develop a new family of products with similar dimensions and constraints, but nowadays, the capacity of reconfiguration has become a major issue for improving the functioning of industrial processes (Revilla, 2008). Indeed, today a main objective is to adapt quickly in order to start a new production or to react in a failure occurrence. Intelligent manufacturing systems (IMS) offer not only both flexibility and reconfigurability, but also this concept brings more than a few ideas of software intelligence

\* Corresponding author. Tel.: +  
E-mail: almd@fe.up.pt (F. Almeida)

meanings, which contemplated characteristics such as autonomy, decentralization, flexibility, reliability, efficiency, learning, and self-regeneration (Revilla, 2008; Mekid, 2009; Shen, 2006).

The manufacturing enterprises of the 21st century are in an environment where markets are frequently shifting, new technologies are continuously emerging, and competition is globally increasing. Manufacturing strategies should therefore shift to support global competitiveness, new product innovation and customization, and rapid market responsiveness (Prajogo, 2007). The next generation manufacturing systems will thus be more strongly time-oriented (or highly responsive), while still focusing on cost and quality. Such manufacturing systems will need to satisfy a number of fundamental requirements, including (Shen, 2006; Chituc, 2009):

- Full integration of heterogeneous software and hardware systems within an enterprise, a virtual enterprise, or across a supply chain;
- Open system architecture to accommodate new subsystems (software or hardware) or dismantle existing subsystems “on the fly”;
- Efficient and effective communication and cooperation among departments within an enterprise and among enterprises;
- Embodiment of human factors into manufacturing systems;
- Quick response to external order changes and unexpected disturbances from both internal and external manufacturing environments;
- Fault tolerance both at the system level and at the subsystem level so as to detect and recover from system failures and minimize their impacts on the workflow environment.

### *1.2. The XPRESS approach*

The EU project XPRESS (IP026674-2) aims at developing a concept of an IMS and intends to establish a breakthrough for the factory of the future with a new flexible assembly and manufacturing concept based on the generic idea of “specialized intelligent process units” (“Manufactrons”) integrated in cross-sectoral learning networks for a customized production and a flexible system organization. This knowledge based concept integrated the complete process chain, from the production planning to the assembly, the quality assurance of the produced/assembled products and the reusability of process units (Peschl, 2010).

The new concept of manufactronic networked factory is developed and demonstrated by a strong industry-lead partnership in order to meet the still remaining industrial needs with regard to:

- Production configuration and simulation – XPRESS intends to significantly decrease the ramp-up time for assembly lines, increase the reusability of assembly components and optimize the entire of the assembly process;
- Manufactron guided production flow – for the assembly and manufacturing of different types and variable volumes of products on a single flexible line and achievement of a high level of reusability;
- Manufactronic machines and human integration – a) reducing the effort needed for setting up a single process; b) providing most efficient and reliable inline quality assurance systems for the processes; c) reacting intelligently on disturbances; d) providing a factory-wide process monitoring system; e) allowing the reuse of disassembled components.

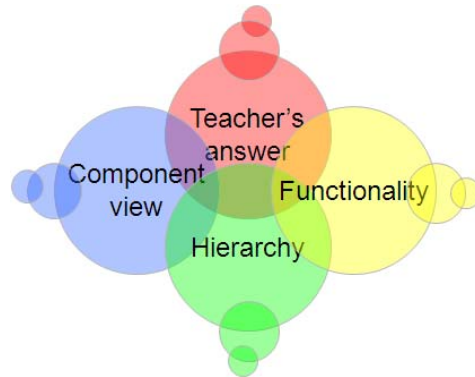
The work report in this paper establishes a breakthrough for the factory of the future with a new flexible assembly and manufacturing concept based on the generic idea of “Manufactron” integrated in cross-sectoral learning networks for a customized production and flexible system organization.

The remainder of this paper is organized as follows: In section 2 we present the standard structure of a manufactron. Section 3 describes the concept of a manufactronic networked factory giving an

overview of its components. Section 4 describes the implemented approach followed by the project. Section 5 presents the main results obtained by the project, particularly related to the four demonstrated scenarios. Finally, the conclusion of our work is drawn and an outlook for further work is given in section 6.

## 2. The manufactron concept

A Manufactron is a self-contained entity, which is encapsulating expertise and functionality and interacts with its environment by the exchange of standardized synchronous messages. This notion of Manufactron can be better understood looking for the four different views presented in Fig. 1.



**Fig. 1.** Different perspectives for manufactron definition

The component view lists several components, which shall be part of every “typical” manufactron. These components can be implemented into a library, the “Manufactronic framework”, in order to re-use the same components for nearly every manufactron. Nonetheless, this is not mandatory. If a manufactron realizes its own components, which are only behaving in the same way, it will comply also to the definition of an “Manufactron”.

The functionality view gives an answer, which functionality has to be realized by a piece of software or order to name it “Manufactron”. Therefore again, the “Manufactron” may rely to its own implementation, if only it’s realizing the needed functionality to be called a “Manufactron”.

The hierarchy view proposes a set of three different levels (Production configuration manufactrons, workflow/quality manager manufactrons and production manufactrons), on which artifacts of the XPRESS project shall be realized. Every manufactron shall fit into exactly one of these levels, where the first and second do have some special restrictions and responsibility. It will be therefore expected, that most of custom-implemented manufactrons will reside on the level of “Production Manufactrons”.

The manufactron shall be self-contained. It is expected that a typical manufactron may be added to a manufactronic factory by just plugging an additional device into the factory’s network. Therefore, the manufactron shall be realized as an independent piece of implementation rather than a very distributed entity, where a lot of different fractions of the entity are to be integrated into different systems of the factory, as to be the enterprise resource planning (ERP) and the manufacturing execution systems (MES) system of different kinds of programmable logic controller (PLC) systems (Ribeiro, 2010).

The manufactron shall not only realize a simple functionality, but shall also provide expertise on this functionality to the outer world. This allows the outer world to state a task to be fulfilled to the manufactron without the need to know about every small detail associated with these tasks. The encapsulation of expertise is therefore the answer to demands stated by multi-variant production (higher levels do not have to concern about small details) and flexibility in terms of production

resources (a task is not depending on a very special welding machine, but can be understood by every welding machine).

The manufactrons are agents that decide how to reach their given goals best, but not when to do it. The task execution is triggered from outside as defined by another manufactron category, named “workflow manager” overlooking the factory level with dedicated knowledge expertise (Almeida, 2010). This results in a manufactron hierarchy:

- Field level: “Production manufactrons” (executing basic manufacturing tasks) and “Super manufactrons” (co-ordinating groups of Production manufactrons);
- Factory level: “Workflow managers” (controlling the production flow of an item) conforming the manufacturing execution system up to production planning;
- Bureau level: “Configuration manufactrons” responsible for finding an optimum production configuration and for the creation of workflow managers for different product variants or for varying production conditions.

The capabilities of a manufactron are described in the manufactron self description (MSD) document. Each manufactron or other entity in the manufactronic factory can request the MSD of a manufactron. The main information contained in a MSD file include the information on the capabilities of the manufactron, the information regarding the task description, and the quality result items generated by the Manufactron after the execution of a task.

### **3. Manufactronic networked factory**

A high challenge of the XPRESS specification and development work is the interaction of the different components of the whole system. The communication scheme between components of the different layers (ERP, shop floor and cell level) and also within the layers must be powerful, flexible and extensible. A main focus of the specification in this area was to develop a uniform and standardised communication protocol for the manufactronic framework. For that purpose, a XML based approach has been chosen, which guarantee a very flexible and extensible system, being at the same time powerful enough to handle all data and signals to be transported between system components.

The basic approach of the manufactronic communication scheme is a synchronous exchange of documents. For that, only two types of documents do exist:

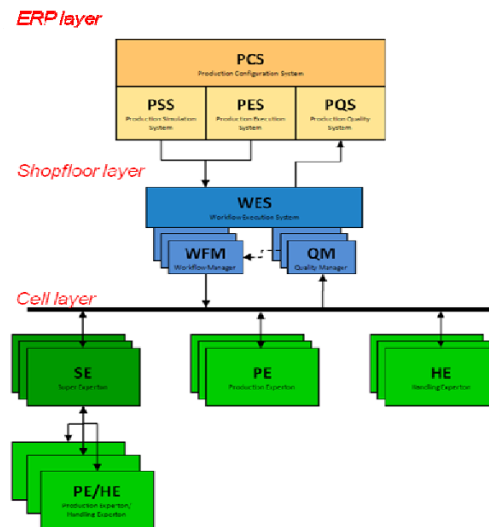
- Task description documents (TDD);
- Quality result documents (QRD).

TDDs provide input information for a manufactron. This document includes all information needed by the experton to perform a task. This includes the information, what to be done, the task goals as well as specific boundary conditions for task performing (Pollak, 2010). The information in the TDD is a XML-based language and has hierarchical structure. On the other side, QRDs are released by the expertons after they received a TDD and performed the task. QRDs do not only contain quality information (as the name might suggest). It contains any kind of data, which is the result of performing a task. The network topology of the manufactronic networked factory is presented in the sections below.

#### *3.1. Production configuration system*

The production configuration system (PCS) is the component responsible for the simulation process, execution start and execution workflow management. During the simulation process or planning phase, its core tasks include the definition of the optimal configurations based on product’s definition, processes and production goals. After finding the best production configurations, the PCS is able to issue production orders by instantiating Workflow Managers, which control all the production

process in the lower level layers. This is called the production phase. If a problem occurs during this phase, the PCS is able to find a sub-optimal configuration to be applied to the production process. Fig. 2 presents the hierarchy of the complete system deployed on the factory.



**Fig. 2.** Overview of the manufactronic architecture (Almeida, 2010)

The system is comprised of the PCS, which is the main subject of this document, the workflow execution system (WES), and the lower level manufactrons: Super Manufactron, Production Manufactron, Human Manufactron and Handling Manufactron. The WES, instantiated by the PCS during the simulation phase or production phase, is comprised of workflow manager (WFM) and quality manager (QM) components. This component, the WES, is the mediator between the PCS and all the other production manufactrons (PMs) or handling manufactrons (HMs) or super manufactrons (SMs). Each started instance of WFM or QM is responsible for the control and organization of the manufactrons underneath it. This allows the WES to suspend or to persist the manufactrons, if no activity is to be performed. It is the responsibility of every manufactron to communicate with dependent or superior manufactrons (SMs or WES “manufactron”). As far as the communication goes, it is done along with the arrows depicted in the figure, representing the exchange of XML data within the system. The system’s communication is synchronous, therefore, each TDD sent to a manufactron must return a QRD. In case that the operation is not performed, a QRD containing an error message must be sent to upper level.

The PCS is divided in three components: production simulation system (PSS), production execution system (PES), and finally production quality system (PQS). Each sub-component has its own components, in order to make PCS implementation easier to maintain. The PSS performs simulation tasks, using different workflows with various production manufactrons and configurations. On the other hand, the PES is responsible for receiving and selecting the best configuration from production jobs issued by external ordering systems, such as SAP. Regarding PQS, this component is responsible for storing and retrieving the quality results in XML formatted files denominated quality result documents (QRDs), which are generated at the end of the production cycle and contain the complete quality information of the entire production process and the product itself.

### 3.2. Distributed workflow execution system

Originally the manufactronic system specification supports only a single workflow execution system (WES). This initial limitation introduced some disadvantages, turning impossible the support for parallelism on lower levels. In fact, manufactrons that received a TDD are required to finish their task and answer with a QRD, before the next TDD can be sent. While this synchronous behaviour reduces system complexity, it prevents simple implementations for pipelined machines. Pipelined machines

can start production of a second product, before the first product is finished. Depending on the size of the pipeline,  $n$  products can be started during the production time of a product.

To mitigate these disadvantages, the concept of a “distributed WES” is introduced. The central factory WES can optionally be assisted by one or more local Sub-WES systems. The Sub-WES can be integrated as part of a machine (hence the term “local”). Its task is to execute workflows locally. Fig. 3 illustrates the distributed WES approach.

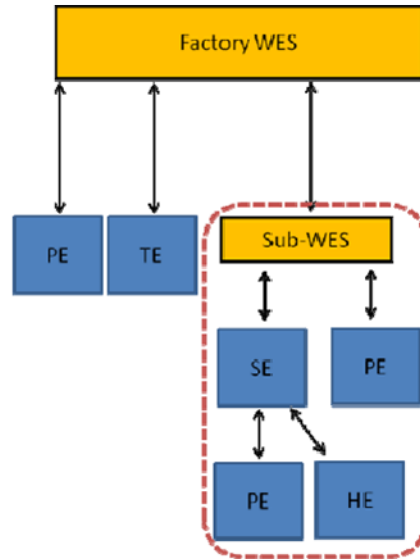


Fig. 3. Distributed WES

One property of the WES is that it can keep track of multiple workflows concurrently, by instantiating workflow managers for each of them. This property solves the parallelism problem in pipelined machines, as the Sub-WES can instantiate a WFM for each product in the pipeline.

Because the Sub-WES is dedicated to a single machine, its workload is more predictable, and communication links between manufactrons and Sub-WES remain local. The delay that is introduced by the WES is therefore much more predictable. Up to a certain extend it is even controllable, by selecting computing and communication hardware to match the machine’s required performance.

Furthermore, the Sub-WES contributes to the robustness of the system. If the Factory WES is unable to issue TDDs, or if the communication infrastructure to the machine fails, the Sub-WES can be instructed to locally re-issue the last TDD(s) repeatedly. This way a fall-back option is created, the machine can continue producing, even when it is offline.

### 3.3. Directory service

The directory service (DS) is a required component in the manufactronic communication framework. It has a supporting role in all communication transactions between the manufactronic components. The DS provides services to register and resolve network addresses and manufactron names. Furthermore, it provides authentication and security services to the communicating parties.

The DS is not a manufactron and has a special interface to be called. The existence of this component brings relevant advantages to the manufactronic networked architecture in terms of robustness, tolerance of intermittent network errors, fast reaction to failures and a reliable messaging system.

The DS stores every change in a persistent storage using XML. The manufactrons are identified by a unique name and a global unique identifier (GUID). Besides that, DS has a ping process, which in regular intervals makes sure whether the registered manufactrons are alive. After directory service starts, it reads data from the persistent storage (if not exists creates an initial repository). It registers all the manufactrons in the ping process (regardless if the status is ALIVE or UNAVAILABLE) and

starts the process. If a node does not answer to a Ping request or a different manufactron answers from the registered endpoint, it is automatically tagged as UNAVAILABLE. An UNAVAILABLE manufactron is removed from the DS after a configurable tolerance time. On the other side, if a node answers to a Ping request and its STATUS was UNAVAILABLE, it is tagged as ALIVE back again.

Fig. 4 depicts the directory service interface of the service.

```

c:\Projects\Xpress\workspace2008\XpressDirectoryService\XpressDirectoryService\bin\Debug\Xpres...
Directory Service Started on http://localhost:6001/DirectoryService
c:\Projects\Xpress\workspace2008\Misc\configurations@Default.xml
RegisterNode <, PES, f1c48712-b374-49eb-a51c-d5367579bd86>
RegisterNode <, wes, ce01839c-ff03-4945-aceb-a11eeec63383>
RegisterNode <, WES Monitor, ab1d0afb-b22e-4eab-a5f4-ec9cf081c5c9>

```

**Fig. 4.** Directory service interface

### 3.4. Monitoring service

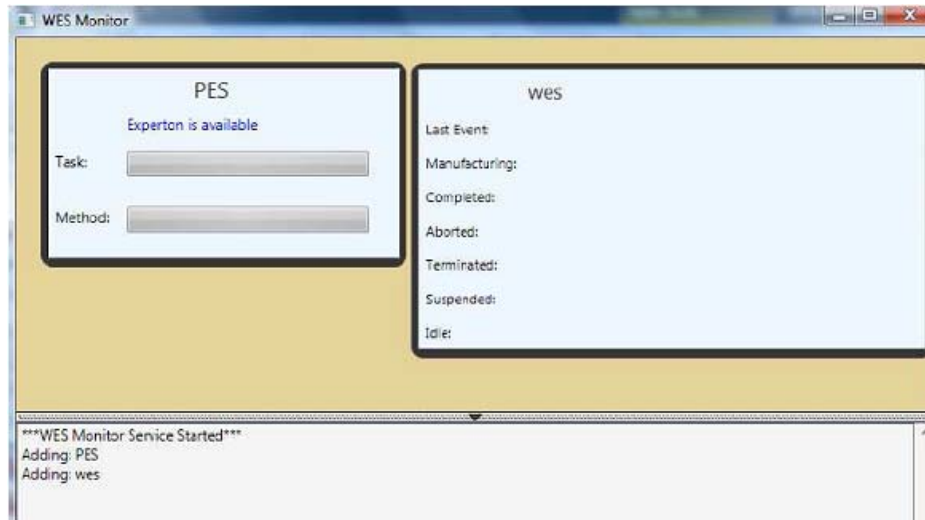
Monitoring service (MS) is a kind of supervisory control and data acquisition (SCADA) service, which is intended to show an overview of the manufactronic factory. It dynamically displays the so-called “widgets”, which is maintained by individual manufactrons. MS uses windows presentation foundation (WPF) as a user interface technology. WPF is an XML-based language, which makes it very suitable to realize a SCADA-like system.

MS puts additional graphical elements to these widgets such as tracking products. Every product has a unique id, such as RFID or barcode during the production, so it can display the whereabouts of the products. MS service has a logging facility, which can show what is happening in the factory. Analyzing this log can provide valuable information to eliminate network errors.

MS is tightly integrated with the directory service. The registration of manufactrons in the monitoring service is completely automatic. MS monitors DS for changes in the manufactronic hierarchy. This is based on Manufactrons’ status, created and updated time. If a manufactron is temporary unavailable (e.g.: intermittent network failure) the widget’s border becomes red on the MS canvas. After the manufactron is removed from the DS, it is removed from the MS as well.

When MS realizes that a new manufactron is registered in the DS, it sends a subscription request directly to the manufactron. The manufactron registers this in its local subscription list, sends a widget template and the initial data to the MS. The manufactron appears immediately on the MS canvas. From now on the manufactron notifies MS of every changes of its status. Although the communication is not real-time, it is close to it. The notification messages frequency can be very high, so it can happen, that the messages arrive in a different order, than they were sent. To solve this, MS just drop those messages, which were sent earlier, than the last received message. A sequence number by manufactron intends to handle this issue. Besides that, as the monitoring service is also a manufactron, it is capable of intervening the execution of the workflow, such as terminating the execution and dropping the product. Although this service only displays the widgets at the moment, it has the potential to become a more powerful controller.

Fig. 5 depicts the monitoring service interface of the service.



**Fig. 5.** Monitoring service interface

### 3.5. “Factory Floor” manufactrons

“Factory floor” manufactrons are the manufactrons that can be found on the factory floor, like the production manufactrons, handling manufactrons, transport manufactrons and Sub-WES.

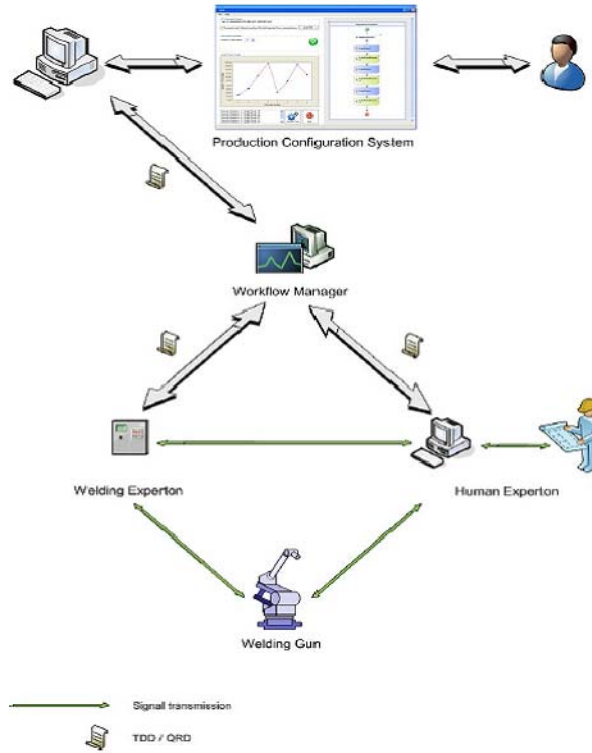
The production manufactron is responsible to perform a task at the shop-floor and implements process knowledge and/or connections to the filed level. Handling manufactron is a special case of a production manufactron that is responsible to handily manipulate a work-piece. Transport manufactron is responsible to transport a work-piece on a factory floor (the XPRESS supports two kinds of transportations: based on conveyor pallets and AGVs). Finally, the sub-WES acts as a unit coordinator realizing the workflow and quality manager attached to a single product.

In most, if not all cases, the manufactron will be communicating with its associated production system, like a PLC system, a weld controller, a robot system or the controls of a vehicle. This communication may be based on 100BASE-TX Ethernet, but other standards or proprietary interfaces are also allowed. The availability requirements for these manufactrons are less demanding, compared to the PCS/PQS, WES and DS, as a failure of one of these components will not lead to a standstill of the complete factory. The amount of processing power needed is greatly dependent on the type of manufactron and its implementation. If processing power allows, it is possible and allowed to run multiple manufactrons on one piece of hardware.

### 3.6. Human-machine interface

Fig. 6 illustrates the production execution system (PES) in its diagram form, where the workflow manager (WFM) object is instantiated through the workflow execution system interface by issuing a TDD, which is forwarded by the WFM to a human handling manufactron and, simultaneously, to a welding manufactron. Both manufactrons together perform a row spot welding task on a car door. The generated quality data is sent back to the quality manager of the WFM, in QRD format. This quality manager assesses the overall quality of each task and their combination and then sends it back to the PCS, where the quality results are displayed to the end user.

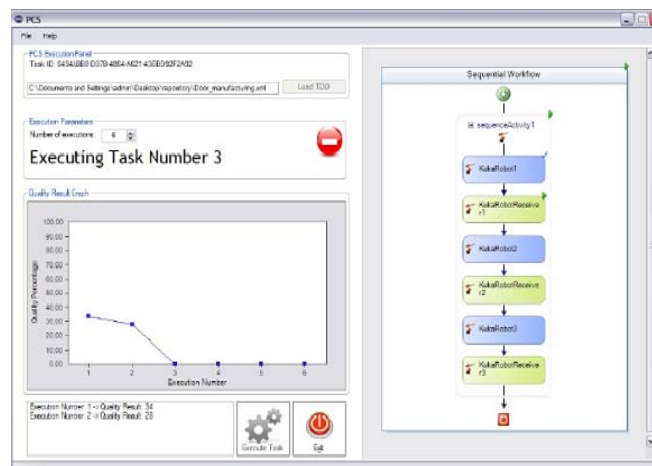




**Fig. 6.** Production execution system diagram

The PES provides a graphical user interface (GUI) that simplifies the end user’s interaction with the available PES functionalities. Among all the available functionalities it is worth to note the loading of XML files with TDD/QRD library, generation of workflow managers (WFM) and quality managers (QM), interface to WES and displaying quality results.

At start-up, the end user is offered an interface where it is possible to load a specific TDD and set the number of executions for the chosen task. After the user starts the PCS execution, a workflow manager (WFM) object is instantiated and the loaded TDD is forwarded to this new object. This object will handle the task description to the lower level manufactory which will perform the task described in the TDD, while the WFM is controlling the lower level manufactory layer by updating the workflow status of each activity. The GUI is able to show this process at run-time. This situation is illustrated in Fig. 7.



**Fig. 7.** PCS GUI working

At the same time, the GUI is able to present the quality results, sent back from the WFM to the PES. These results are presented to the end user in a graphical form where the X-axis represents the execution number and the Y-axis represents the quality percentage obtained. After the execution phase, the graphic will contain all the quality results from all the executions and the workflow viewer will display all the activities as finished.

### 3.7. *Interface to external simulation tools*

The PSS has two possibilities to access data from outside its own area of responsibility: from the PCS knowledge base and from an external simulation tool. The interface to the external simulation tool will be realized via a “simulation manufactron”.

The simulation manufactron is based on the universal manufactron and therefore presents to the PSS the I/O interface layer of the universal manufactron. When the PSS requires the services of an external simulation tool, it sends a TDD to the simulation manufactron and gets a QRD in return. The details of unpacking data from the TDD, sending it to the simulation tool, receiving the results of the simulation and packing them into a QRD are all hidden behind the manufactron I/O interface.

Using this approach, the knowledge about how to interpret the TDD data is encapsulated in the simulation manufactron. This encapsulation provides the benefit that any change to the TDD structure is limited in scope. Without it, every time the TDD structure is modified the simulation tool would have to be reprogrammed to understand the new way of data representation.

## 4. **Implementation**

### 4.1. *Workflow manager*

The workflow manager is a simple console application, with three services to host: Manufactronic service, workflow runtime and workflow communication service. The workflow runtime hosts two additional services: tracking and persistence service, which are based upon the standard SQL server implementation of the windows workflow foundation (WWF). The workflow manager can simultaneously execute several tasks.

The workflow manager provides an additional windows communication foundation (WCF) service, composed by the following methods:

- **ValidateTask** – turns possible the validation of a TDD before the execution;
- **GetAllWfStatus** – get the status of a workflow and its result is given in XML format;
- **GetInstances** – returns all instances according to the filter, which can have one of the following values: running, completed or all;
- **RaiseWorldEvent** – provides an external interception possibility in the execution of the workflow.

There is also a Workflow Monitoring application, which is an ASP.NET web site. This application communicates with the WfmQm through the workflow communication service and it has read access to the workflow tracking and persistence database.

On the website it is possible to check the running instances, the quality results of the executed tasks and the tracking information (when, which task has been executed, with what result) of the completed and running workflows. Furthermore, on the site it is possible to intercept the process of an execution. For example, a WorldEvent can be sent to the WfmQM or a workflow can be aborted if it has a deadlock or an infinite cycle. Fig. 8 depicts an execution example of the workflow monitoring application.

[Home](#)

### Available workflows

Conditional1.xml	<a href="#">validate</a>
Door_manufacturing.xml	<a href="#">validate</a>
Performance.xml	<a href="#">validate</a>
State1.xml	<a href="#">validate</a>

[show\\_all](#) [show\\_running](#) [show\\_completed](#)

[448a2a93-7392-4fbe-a554-dbe619055f4d \(Completed\)](#)

**Status:**

Last event: Completed  
Overall Quality: 11

**Tracking Data**

2008-05-20T19:18:10.51+02:00 a1 sending to t2 (tdd: E6AAD465-943B-4cd6-8832-D917C46345D9)  
2008-05-20T19:18:12.133+02:00 a1 has finished (q: 32)  
2008-05-20T19:18:12.143+02:00 a2 sending to t1 (tdd: C21A9F10-7663-444a-A7EE-39264A60EAFA)  
2008-05-20T19:18:13.07+02:00 a2 has finished (q: 11)  
2008-05-20T19:18:13.103+02:00 a3 sending to t2 (tdd: C662308C-DFC7-4f5b-967A-DFFA91EE94DF)  
2008-05-20T19:18:13.26+02:00 a3 has finished (q: 41)

Fig. 8. Example of the workflow monitoring application

#### 4.2. Workflow manager template

The workflow manager template is embedded into a task description document (TDD). In the manufacturing hierarchy every “instruction” is a TDD. At Workflow, TDD contains one main task, which has the workflow control-flow (executable program) and additional embedded TDDs identified by a TddId, which the control-flow sends to the underlying manufactrons. It is important to emphasize that the TDD is a unique product instance, which follows the rules of the WFM template.

Fig. 9 gives an example of a sample control-flow.

```
<cf.ControlFlow>
  <cf.SequenceActivity>
    <cf.ListenEVT Name="polish_listen" ActivityId="evt1" Type="detected" Content="from=polish"/>
    <cf.EVTReceived Name="product_available_1" ActivityId="evt1"/>
    <cf.SendTDD Name="do_polishing" TddId="E6AAD465 943B 4cd6 8832 " ActivityId="a1" Recipient="polish"/>
    <cf.QRDReceived Name="polish_ended" ActivityId="a1"/>
    <cf.IfElseActivity>
      <cf.IfElseBranchActivity Name="manu_welding">
        <cf.IfElseBranchActivity.Condition>
          <cf.RuleCondition Condition="!IsAvailable(&quot;rsw2&quot;) || IsAvailable(&quot;rsw1&quot;)" />
        </cf.IfElseBranchActivity.Condition>
        <cf.ListenEVT Name="manu_welding_listen" ActivityId="evt2" Type="detected" Content="from=rsw1"/>
        <cf.EVTReceived Name="product_available_2" ActivityId="evt2"/>
        <cf.SendTDD Name="do_manu_welding" TddId="C21A9F10-7663-444a-A7E" ActivityId="a3" Recipient="rsw1"/>
        <cf.QRDReceived Name="manu_welding_ended" ActivityId="a3"/>
      </cf.IfElseBranchActivity>
      <cf.IfElseBranchActivity Name="auto_welding">
        <cf.ListenEVT Name="auto_welding_listen" ActivityId="ev3" Type="detected" Content="from=rsw2"/>
        <cf.EVTReceived Name="product_available_3" ActivityId="evt3"/>
        <cf.SendTDD Name="do_auto_welding" TddId="C662308C-DFC7-4f5b-967A-" ActivityId="a4" Recipient="rsw2"/>
        <cf.QRDReceived Name="auto_welding_ended" ActivityId="a4"/>
      </cf.IfElseBranchActivity>
    </cf.IfElseActivity>
  </cf.SequenceActivity>
</cf.ControlFlow>
```

Fig. 9. Sample control-flow of the workflow manager template

The *cf:ControlFlow* is always the root and contains one of the two main containers (Sequence and State). The containers contains compound and simple activities, which can be standard WF activities and manufactronic primitives too. The template is written in a special manufactronic dialect, but it is similar to extensible object markup language (XOML) as much as it can. In the following, sections defining the primitives and their corresponding XOML variant will be presented.

The sequence container contains a sequence of activities. It is important to mention that every workflow must have an entry and exit point. In the state container the *InitialState* and the *CompletedState* exists and only one state can be activated at a time. The states contain an initialization sequence and an event driven activity. The initialization sequence is executed, when the workflow entries into a state activity and at the end it waits for an event, which can trigger the workflow to proceed to a next state. The next state to follow is defined in the *SetStateActivity*. When the *CompletedState* is activated the workflow terminates. State machine's path of execution is arbitrary according to the order of events and data. Every execution can differ, contrary to the sequence container, where the execution path is determined beforehand.

Fig. 10 defines the template for state container.

```
<cf:StateContainer InitialStateName="sInitial" CompletedStateName="sEnd">
  <cf:StateActivity Name="sInitial">...</cf:StateActivity>
  <cf:StateActivity Name="sSecond">...</cf:StateActivity>
  <cf:StateActivity Name="sEnd"/>
</cf:StateContainer>

<wfa:XpressWfStateContainer xmlns:wf="http://schemas.microsoft.com/winfx/2006/xaml/workflow"
xmlns:wfa="http://www.xpress.org/Wf/Activities" x:Name="Container" TaskDefinitions="{x:Null}"
InitialStateName="sInitial" DynamicUpdateCondition="{x:Null}" CompletedStateName="sEnd">
  <wf:StateActivity x:Name="sInitial">
    <wf:StateInitializationActivity x:Name="sInitial">...</wf:StateInitializationActivity>
    <wf:EventDrivenActivity x:Name="edInitial">...</wf:EventDrivenActivity>
  </wf:StateActivity>
  ...
  <wf:StateActivity x:Name="sEnd" />
</wfa:XpressWfStateContainer>
```

**Fig. 10.** Template definition for state container

The template includes several workflow primitives, respectively:

- Sequence activity – can contain sequence of activities, which are executed one-by-one. If the execution stops, for example waiting for an event, the workflow won't proceed to the next step;
- Parallel activity – can contain multiple threads. The threads run pseudo-parallel, which means that only one activity is executed at a time, but if one thread is blocked the others can proceed freely. It is similar how one processor can run multiple threads in modern operating systems;
- List event – notifies the Workflow Runtime that the workflow is waiting for an event. When this event is received by the Workflow Runtime, the corresponding *EVTReceived* activity is triggered;
- Send event – sends an event to a manufactron. It can be paired with an *EVTReceived*, but it is not mandatory;
- Event received – this activity is waiting for an event from the Workflow Runtime;
- Send TDD – sends a TDD to a manufactron. This activity always has a corresponding *QRDReceived* activity, because it is a requirement by the manufactronic system;
- QRD received – this activity is waiting for an event from the Workflow Runtime;
- If-else-activity – evaluates a *RuleCondition* and decides which branch to execute. In this example it checks the availability of manufactrons;

- While activity – executes the *SequenceActivity* in the *WhileActivity*'s body until the *RuleCondition* evaluates to true;
- Generate *SendTDD* activities – the transformation substitutes the activity with Num pieces of *SendTDD* activity. It is used for measure the workflow execution system's performance;
- Delay activity – the execution is delayed with *TimeoutDuration*.

## 5. Results

The manufactronic networked approach and all the production manufactrons developed and their collaboration were tested to demonstrate their functionality. The existence of several demonstration scenarios encourages potential suppliers to provide their equipment based on the manufactronic concept. Additionally, potential end users have the possibility to see the manufactronic networked factory running and can therefore be convinced in an easier way of the manufactronic concept and its advantages.

The following four demonstrators were considered:

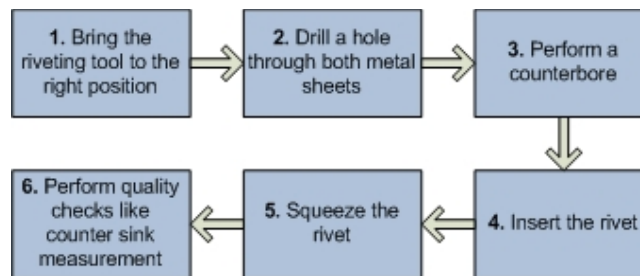
- Demonstrator #1: Quality inspection and process monitoring as well as worker assistance in aeronautic industry;
- Demonstrator #2: Planning process and automatic robot path generation in automotive industry;
- Demonstration #3: Worker guidance and worker behaviour interpretation in automotive industry;
- Demonstration #4: Highly flexible and multi-variant production in electrical industry.

### 5.1. Demonstration #1

This demonstrator is the only one which is directly integrated into an existing and running production line. For that reason, a smooth integration without hampering or slowing down the production is required. The demonstrator intends to fulfil the following objectives:

- Demonstration of the abilities of the riveting manufactron;
- Demonstration of the reliability of the quality assurance system;
- Demonstration of closed quality loops for real-time parameter adaptation.

Materials of the panels are aluminium and titanium sheets having different thickness. Due to the fact that the demonstrator is completely integrated into a running production line, real panels of an aircraft are used. The costs of one panel or hampering the production are very significant (estimated between 100.000€ and 500.000€), therefore, the integration of the system into the production line has to be done very carefully). For setting one rivet, several processes are performed. The usual sequence of setting a rivet is illustrated in Fig. 11.



**Fig. 11.** Sequence of setting a rivet

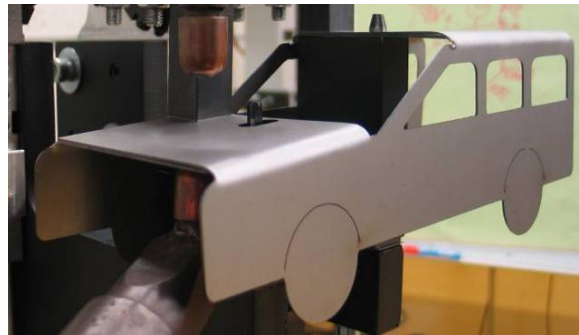
The demonstrator #1 provided the following results:

- It demonstrated the 100% quality assurance of production processes by embedding quality assessment software for the riveting process;
- It demonstrated a reactive production with closed-loop control sequences, and the flexible and fault-tolerance reaction by the dynamic adaption of process parameters based on the quality assessment;
- It demonstrated the feedback of quality information to CAD data by the visual representation of quality information in virtual CAD environments;
- It demonstrated the feasibility of the Manufactronic approach in the aeronautics sector.

### 5.2. *Demonstration #2*

This scenario demonstrates the cooperation of a handling manufactron and a welding manufactron within an application in the automotive industry. The focus of this scenario is the demonstration of the capabilities of the handling manufactron in path planning, automatic path generation and quality assurance. Besides that, this scenario intends to demonstrate the product tracking and production data feedback gathering by the workflow managers.

The scenario consists of three different cars types (station wagon, sedan and coupe) having different shapes and Fig 12 shows a station wagon.



**Fig. 12.** Station wagon

Each product type is built of two metal sheets (left and right side of the car frame). The material and the thickness of the metal sheets do not differ from each other. To weld the different product types, a couple of welding spots are needed. The number and position of the spot differ from type to type. For approaching the different spot locations, a welding gun (mounted on a robot) is used. The insertion and removing of the product from the gripper is done manually.

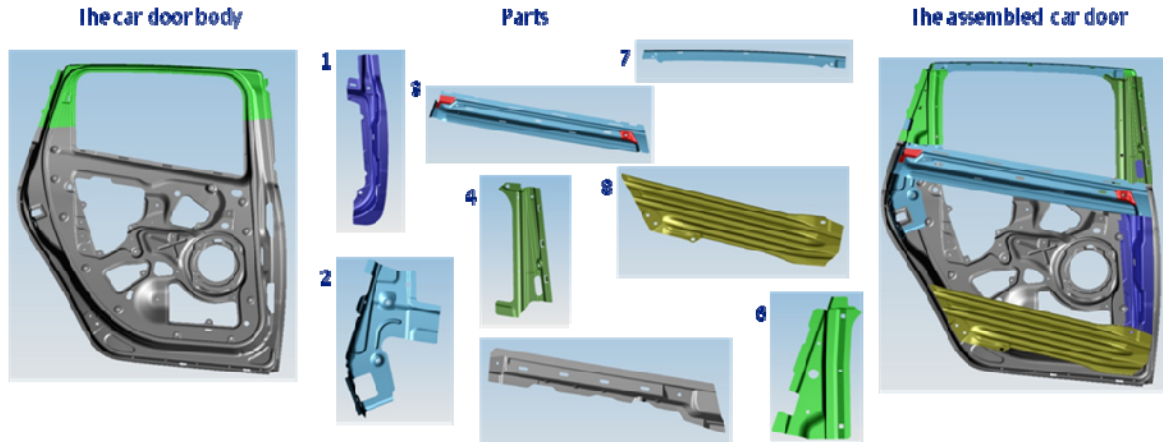
The demonstrator #2 provided the following results:

- It demonstrated the 100% quality assurance of production processes by embedding visual inspection of robot path monitoring;
- It demonstrated a reactive production with closed-loop control sequences and the flexible and fault-tolerance reaction by the semi-automated robot path generation;
- It demonstrated the XPRESS approach for a holistic factory-wide process control and monitoring system by gathering quality data of both welding and handling processes;
- It contributed to decrease of the ramp-up time for the set-up of production line and the optimization of the product cycle time by the semi-automated robot path generation;
- It demonstrated the feasibility of the XRESS concepts in automotive industry.

### 5.3. Demonstration #3

This demonstrator actually has two different setups. The biggest part is the demonstration of the worker integration into the manufactronic concept; another setup is the inclusion of handling manufactron which focuses on the cooperation of two handling manufactrons based on Cornau robots.

For performing the robot scenarios and the monitoring of the worker sequence (in body shell), cars doors are used. Fig. 13 illustrates the production assembly steps of a car door.



**Fig. 13.** Assembly process of a car door

It is relevant to mention that the materials used in those scenarios are not relevant, because the scenarios do not depend on the material properties. In addition, the processes (in terms of joining processes) are not that relevant in those scenarios.

The worker integration scenarios provided the following results:

- It demonstrated the 100% quality assurance of production processes by monitoring the correct sequence of handling tasks by humans;
- It demonstrated the reactive production as well as the flexibility and fault-tolerance in production by the identification of wrong components or faulty components using video inspection;
- It demonstrated the potential of the XPRESS concept for factory-wide quality data gathering by gathering and assessing quality data of different tasks;
- It demonstrated the quality data monitoring by feeding back quality information to the human.

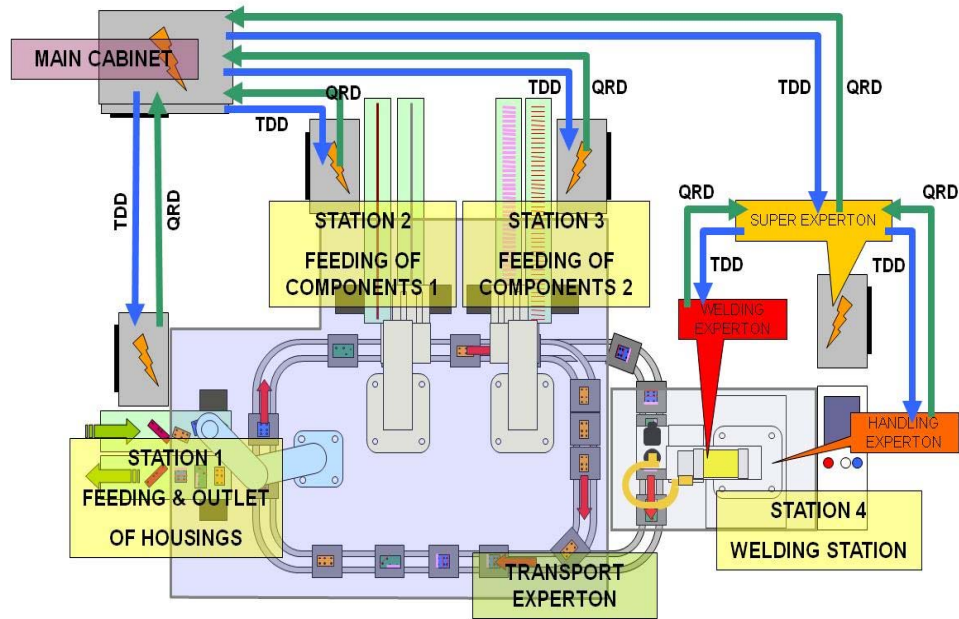
The robot cooperation scenario provided the following results:

- It demonstrated the flexible reaction on unexpected production volumes in case of manual production tasks by showing the exchangeability of tasks done by humans and robots;
- It demonstrated the reusability of assembly equipment by wrapping a Cornau robot with a Handling Manufactron shell developed for a KUKA robot.

### 5.4. Demonstration #4

This scenario demonstrates the cooperation of several manufactrons within a manufactronic machine for electrical industry. This scenario provides a holistic view of the manufactron approach, because the manufactron approach does not only demonstrate with one or two single manufactrons, but each relevant component in the machine is implemented as a manufactron. Thus, a complete manufactronic production is demonstrated.

Fig. 14 illustrates the setup of the machine using a conveyor pallets system.



**Fig. 14.** Setup of the machine

The scenario is equipped with a cyclic transport system and pallets and four stations. On the top of the pallets, the housings are placed. The transport system consists also of a switch for bypassing products of the welding station (station 4). On station 1, the randomly incoming housings are placed on the pallets and the assembled housings are removed. In station 2 and station 3 the components are fed and placed on the housings. Station 4 is equipped with a handling and a welding device. Both are covered by manufactronic shells and are coordinated by a super manufactron. It is also important to mention that the experton and manufactron names are used indiscriminately in Fig. 14.

Three different housings are used in order to simulate three different products/products variants. Each housing has different holes in which two different types of components will be placed. This situation is illustrated in Fig. 15.

**Fig. 15.** Assembly process of components

There are four types of housings. However, only three are used for production. If the fourth (violet) housing is placed on the conveyor belt, the video system has to detect it as a faulty housing and must trigger an event to reject it immediately. Besides that, one housing (yellow) is vertically equipped with components only. For that reason, yellow housings are bypassed at station 4.



The demonstrator #4 provided the following results:

- It demonstrated the decrease of changeover time needed for new product variants by the flexible handling of task description documents (TDD) of the workflow execution system (WES);
- It demonstrated the flexible reaction on production volumes by the dynamic routing capability of the WES;
- It demonstrated the reusability of assembly equipment by extending several low-end devices with a manufactronic shell;
- It demonstrated the reactive production as well as the flexibility and fault-tolerance in production by the dynamic routing capabilities of the WES;
- It demonstrated the potential of the XPRESS concept for factory-wide quality data gathering by gathering and assessing quality data of different tasks;
- It demonstrated the feasibility of the XPRESS concept in electrical industry.

## 6. Conclusions

XPRESS meets the challenge to integrate intelligence and flexibility at the “highest” level of the production control system as well as the “lowest” level of the singular machine. The XPRESS manufacturing system integrates a superior cost-efficient production configuration tool in which a complete production line can be reliably simulated as a digital factory. In fact, XPRESS shifts the whole production process from a resource-intensive industry towards knowledge-based and customer-driven approach.

XPRESS allows the implementation of a multi-variant system making possible to have an adequate number of production lines for the manufacturing of adequate quantities of respective goods using and adequate number of manufactrons in order to meet the requirements of increasing product variants and producing at ever smaller lot sizes. Due to the knowledge and responsibility segregation within the XPRESS system, the various production units are easily extendable and exchangeable and thus offer an unlimited “plug & produce” functionality. Different product variants can be produced with the same assembly units (Manufactrons) on the same production line. The new manufactronic concept achieves a high level of reusability of assembly equipments and is fast, flexible, reconfigurable, and modular.

The radical innovations of the “Manufactronic Networked Factory” are knowledge and responsibility segregation and trans-sectoral process learning in specialist knowledge networks. Assembly units composed of Manufactrons can flexibly perform varying types of complex tasks, whereas today this is limited to a few pre-defined tasks. By sharing the specific knowledge of each manufactron in a network, other manufactrons are able to learn from each other in one production line, but also between different lines as well as different production units. This architecture allows continuous process improvement. Therefore, XPRESS is able to anticipate and to respond to rapidly changing consumer needs, producing high-quality products in adequate quantities while reducing costs.

The concept of manufactronic networked factory was demonstrated in three representative applications (automotive, aeronautics and electrical industry). XPRESS realized a reactive production with closed-loop control sequences. With this method it was possible to react more flexibly and fault-tolerantly on disturbances and, therefore, the reliability and availability of the production line was increased. With XPRESS it was possible to reach an availability of up to 92% (state-of-the-art is 87%). An important industrial need is also to have a holistic factory-wide process control and monitoring system. XPRESS addressed this issue and proposed an interoperability concept, in which different hardware and software components can be addressed and connected via standard interfaces, enabling a user-friendly, flexible and reliable production concept and also factory-wide process

controlling and monitoring including weak-point analysis. Feedback to CAD databases in order to optimize the construction of a part is also possible. Finally, the quality assurance system was able to provide a 100% inline non-destructive quality monitoring. Time needed for the destructive tests was reduced drastically and a reduction of the costs of 30%-40% was also reached. Besides that, based in the demonstration scenarios, the ramp-up time for the set-up of production line decreased up to 50% and the changeover time decreased up to 80%.

### Acknowledgments

This work was carried out with the support from the European Commission under the Sixth Framework Programmer for Research and Technological Development, in the Integrated Project XPRESS (IP026674-2).

### References

- Almeida, F., Terra, B., Dias, P. & Gonçalves, G. (2010). Transport with conveyors in the factory of the future. Proceedings of the IEEE International Conference AQTR, Cluj-Napoca, Romania, 3, 1-6.
- Chituc, C. & Restivo, F. (2009). Challenges and trends in distributed manufacturing systems: are wise engineering systems the ultimate answer? Proceedings of the Second International Symposium on Engineering Systems, Massachusetts, USA, 1, 121-136.
- Mekid, S., Pruschek, P. & Hernandez J. (2009). Beyond intelligent manufacturing: a new generation of flexible intelligent NC machines. *Mech Mach Theory*, 44(1), 466-476.
- Peschl, M. (2010). Executive summary of XPRESS. XPRESS Newsletter, available on [http://www.xpress-project.eu/upload/XPRESS\\_42MonthsPublishableExecutiveSummary.pdf](http://www.xpress-project.eu/upload/XPRESS_42MonthsPublishableExecutiveSummary.pdf) and accessed on 12/05/2011.
- Pollak, J., Sarveniazi, A. & Link, N. (2010). Retrieval of process methods from task descriptions and generalized data representations. *International Journal of Advanced Manufacturing Technology*, 53(5), 1-12.
- Prajogo, D., Laosirihongthong, T., Sohal, A. & Boon, S. (2007). Manufacturing strategies and innovation performance in newly industrialized countries. *Industrial Management & Data Systems*, 107(1), 52-68.
- Revilla, J., Christo, C. & Carreira, C. (2007). Intelligent Manufacturing systems – vision about the future. Proceedings of the IEEE International Symposium on Industrial Electronics, Vigo, Spain, 3, 3209-3214.
- Revilla, J. & Cadena, M. (2008). Trends in intelligent manufacturing systems. Proceedings of the World Congress on Engineering, London, UK, 2, 1257-1262.
- Ribeiro, T. & Gonçalves, G. (2010). Formal methods for reconfigurable assembly systems. Proceedings of 15<sup>th</sup> International Conference on Emerging Technologies and Factory Automation, Bilbao, Spain, 1, 323-329.
- Sanchez, L. & Nagly, R. (2001). A review of agile manufacturing systems. *International Journal of Production Research*, 39(16), 3561-3600.
- Shen, W. & Norrie, D. (1999). Agent-based systems for intelligent manufacturing: a state-of-the-art survey. *Knowledge and Information Systems (KAIS)*, 1(2), 129-156.
- Shen, W., Hao, Q., Yoon, H. & Norrie, D. (2006). Applications of agent-based systems in intelligent manufacturing: an update review. *Advanced Engineering Informatics*, 20(4), 415-431.
- Yang, H. & Xue, D. (2003). Recent research on developing Web-based manufacturing systems: a review. *International Journal of Production Research*, 41(15), 3601-3629.