

This item was submitted to Loughborough's Institutional Repository (<https://dspace.lboro.ac.uk/>) by the author and is made available under the following Creative Commons Licence conditions.



CC creative commons  
COMMONS DEED

**Attribution-NonCommercial-NoDerivs 2.5**

**You are free:**

- to copy, distribute, display, and perform the work

**Under the following conditions:**

 **Attribution.** You must attribute the work in the manner specified by the author or licensor.

 **Noncommercial.** You may not use this work for commercial purposes.

 **No Derivative Works.** You may not alter, transform, or build upon this work.

- For any reuse or distribution, you must make clear to others the license terms of this work.
- Any of these conditions can be waived if you get permission from the copyright holder.

**Your fair use and other rights are in no way affected by the above.**

This is a human-readable summary of the [Legal Code \(the full license\)](#).

[Disclaimer](#) 

For the full text of this licence, please go to:  
<http://creativecommons.org/licenses/by-nc-nd/2.5/>

---

# RI-MACS: An Innovative Approach for Future Automation Systems

---

Roberto CheccoZZo , Fulvio Rusinà , Leonardo Mangeruca ,  
Andrea Ballarino , Claude Abadie , Robert Harrison , Radmehr  
Monfared\*

roberto.checcoZZo@comau.com – COMAU, Italy

fulvio.rusina@comau.com – COMAU, Italy

leon@parades.rm.cnr.it - PARADES, Italy

a.ballarino@itia.cnr.it - ITIA-CNR, Italy

claudio.abadie@fra.schneider-electric.com – Schneider Electric, Germany

R.Harrison@lboro.ac.uk - Loughborough University, UK

R.P.Monfared@Lboro.ac.uk – Loughborough University, UK

\*Corresponding author

**Abstract:** The present global competitive environment poses formidable challenges to the European manufacturing infrastructure. The cost of installation and set-up of a typical manufacturing plant over its lifetime is typically 30% of its total costs. The reasons for this include the relative inflexible communication infrastructure in the manufacturing process and the difficulty of porting existing application software into new configurations. The European funded RI-MACS (Radically Innovative Mechatronics and Advanced Control Systems) research project was initiated to address some of the challenges in the future automation systems. Its mission is to bring the innovation offered by the Information Technology (IT) domain into the production life cycle, improving some of the existing inflexibilities in typical manufacturing automation plants. In this paper, some of the key aspects of automation systems engineering that require changes to ensure future competitiveness are highlighted. An overview of current approaches to automation is reviewed in relation to wired and wireless networking, automation architectures, and design environments. Key areas of research and innovation within the RI-MACS project are discussed including the runtime architecture of Intelligent Autonomous Mechatronics Components (IAMC) including integrated support for Web services and the integrated design environment with support for digital design and validation.

**Keywords:** automation, web-based, service-oriented architectures, wireless, manufacturing, virtual engineering, distributed systems.

---

## 1 Introduction

The present global competitive environment poses formidable challenges to the European manufacturing infrastructure; factors behind this include: the need for speed and flexibility in responding to continuously changing market demands, manufacturing cost

---

*RI-MACS: An Innovative Approach for Future Automation Systems*

reduction, improved time-to-market, lower plant and equipment costs through improved re-use in order to amortize these costs over longer periods.

Realising the flexibility needed to adapt quickly to market needs whilst at the same time minimising equipment, start up and operation costs to ensure the economic health of the enterprise seems an almost impossible task. Advances in manufacturing and information technology have made it possible to begin the journey towards adapting the enterprise to the goals set out above. However, compounding the difficulties to be overcome, in designing the manufacturing plants of the future, is the need to consider the large investments already made in existing plants in legacy technology that must continue to be supported.

Studies with potential end-users of RI-MACS have revealed that the cost of installation and set-up of a typical manufacturing plant over its lifetime is about 30% of its total costs. Maintenance and downtime account for another substantial portion of operating costs. If a plant has to be adapted repeatedly to new products by changing its process flow and introducing new or replacing obsolete/non-competitive equipment (often provided by different vendors), then downtime and installation costs rise considerably. The reasons for this include the relative inflexible communication infrastructure among components of the manufacturing process and the difficulty of porting existing application software into new hardware configurations. Significant costs are also incurred due to the difficulty in implementing changes without stopping the production systems for extended periods. In this regard, there is a widespread need for an open environment to enable the “plug-and-play” operation of automation system components and better integration with digital engineering environments.

European manufacturing industry is facing a strong competitive challenge from countries such as China and India, where labour costs are substantially lower. Thus optimizing manufacturing processes is certainly now far more important than in the past in order to maintain competitiveness and in turn significant levels of employment in manufacturing compatible with the goals of the European Community. As EU industries find it increasingly difficult to compete with lower labour costs countries, it is becoming essential to improve productivity dramatically and to drive manufacturing innovation.

Without radical innovation, European manufacturers will find it increasingly difficult to cope in the future. The RI-MACS (Radically Innovative Mechatronics and Advanced Control Systems) research project aims to address some of the key challenges in future automation systems. The speed of innovation in the Information Technology (IT) domain has been increasing over many years compared with the other domains. Therefore, leveraging this innovation is essential to develop radically innovative manufacturing plants and processes. A trend towards an ever-increasing gap between the advent of new ideas and products in IT and their utilisation in manufacturing plants is recognized by the European industrial and academic research groups. The primary goal of the RI-MACS project is to bridge this gap.

## **2 The vision of RI-MACS**

### **2.1 Introduction**

It is envisaged that radical innovation can and should be achieved in all phases of production life cycle, including engineering, commissioning, operation and maintenance, as described below.

- **Engineering:** At this stage of the production lifecycle substantial potential for improvement is offered by the simulation and validation of automation system and the provision of better support for diagnostics.
- **Commissioning:** The introduction of emerging IT technology could drastically reduce commissioning time and cost through the provision of automatic plant configuration, simulation and monitoring, contributing to a faster ramp-up to volume production
- **Operation:** Improved IT support for machine retooling and reconfiguration could significantly contribute to a reduction in production line cycle times, thereby improving performance.
- **Maintenance:** The introduction of new technologies to enable, cost effective, widespread system monitoring will allow enhanced maintenance prediction, planning and response, thereby dramatically reducing downtime and costs.

To enable radically innovation in the manufacturing process, the RI-MACS project features an number of key areas for innovation. This paper addresses the RI-MACS approach to advanced digital engineering, automation architectures and related quality of service (QoS) issues that need to be addresses in distributed manufacturing systems.

**Advanced Digital Engineering:** although many engineering tools are available today for the design and realization of the control cycle, there is still a lack of an encompassing methodology that seamlessly integrates all such tools to support the design of the production process, from the requirement specification and validation, down to software development and validation. While the design and validation of the machine and process nominal cycles is well understood, the design and validation of diagnostics and fault management software remains a case-by-case process. The RI-MACS project has developed an approach providing automatic support for diagnostic and fault management software generation and support for validation, together with the nominal cycle development. Within RI-MACS new tools have been developed that extend the capabilities of digital engineering within the automation domain beyond virtual commissioning to the virtual design of machine behaviour and its verification against multiple application scenarios.

**Automation Architectures:** the RI-MACS project proposes a control architecture based on the following principles: service encapsulation and an introduction of open communication standards alongside a widespread deployment of Ethernet communication technology combined with strong support for existing PLC technology where appropriate, e.g. low-level IEC 61131 based control logic. The RI-MACS

architecture implemented consists of a defined set of Intelligent Autonomous Mechatronics Components (IAMCs); see section 3.3 for more details.

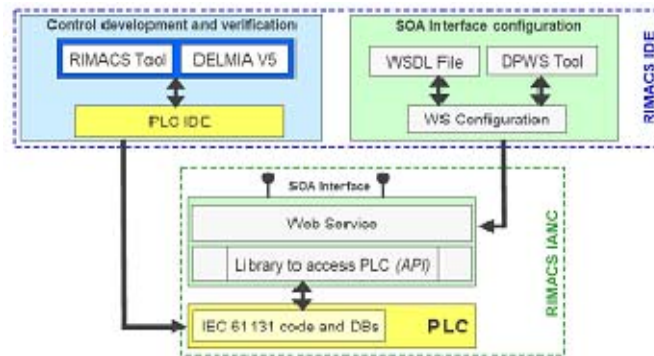
Service encapsulation is implemented via the adoption of a Web services and a service-oriented approach that is expected to play a significant role in future manufacturing plants and processes. Service encapsulation introduces modularity in the control architecture, therefore increasing automation component reuse. The decoupling between service invocation (client side) and service implementation (server side), together with the introduction of open communication standards, ensures that competing solutions from different vendors will be able to integrate and interoperate within the same production lines. Moreover, the adoption of a vendor independent programming model at the process level will allow the process providers to focus on the development of the control applications without wasting costly effort on implementation details. The widespread deployment of the Ethernet communication medium is expected to reduce drastically networking costs, through the utilisation of more standardised Ethernet devices in high volumes, and reduced interconnection costs and effort arising from the current deployment of highly vendor-specific solutions.

**Quality of Service:** The introduction of new networking technologies in the manufacturing process represents an unprecedented opportunity to bring innovation to production plants. The RI-MACS project utilises both wired and wireless communication technologies. A critical aspect of the increased use of both wired and wireless devices within automation systems is the ability to specific and achieve the required quality of service between the various modules of the system. A key aspect of the research has thus been a study of Quality-of-Service (QoS) management across the RIMACS architecture. Other work into QOS has proven to be a valuable support in other domains for streaming applications, e.g., in relation to human communications and entertainment, and are gradually migrating to the Internet as a support tool for applications such as Voice and Video over IP. The RI-MACS research envisages that, as the automation systems change from static, vendor-oriented architectures to more dynamic, open paradigms, QoS management will be an important strategic tool to handle the increasing service management complexity that will quickly occur in such networks. QoS techniques are central to SOA provisioning in business computing applications and are expected to at the root of the establishment of emerging business models of SOA in the manufacturing domain as well. Thus QoS management support has been integrated into the RI-MACS automation architecture; see section 3.5 for more details.

## **2.2 Innovation in Design and Control**

The prototype RI-MACS integrated design and control environments are illustrated schematically in figure 1. The design environment significantly extends the capabilities of Delmia Automation in terms of the innovation in the provision of cycle timing diagrams as a method of specifying machine behaviour and a new methodology for the incremental testing of machines from their individual mechatronic components upwards.

**Figure 1:** RIMACS Integrated Design and Mechatronic runtime environments



The RI-MACS runtime automation architecture features Intelligent Autonomous Mechatronics Components (IAAC). This architecture integrates the traditional strengths of PLCs with the additional capability to encapsulate modules of low-level real-time behaviour within higher-level Web services, enabling a more open, modular approach to automation with readily supports both reconfiguration and seamless integration with higher level IT and business systems. The approach adopted also supports a stepwise migration path to automation devices whereby initially separate web-servers can be utilised, before progressive the integration of web services within real-time control device functionality, ultimately through the creation of mechatronic objects; see figure 2.

### 3 Automation Architectures

#### 3.1 Current Automation Architectures

A number of research efforts have been carried out in recently with the objects of facilitating the introduction of new technologies in automation plants and of harmonizing/unifying the currently fragmented aspects of automation systems management, in particular, those aspects that are characterized by different requirements and to place them into a homogeneous architecture. In this regard, the notable strategies are: the Service-Oriented Architecture approach (IBM 2005, OASIS 2006, Jammes 2005b), the TORERO device-centric approach (Torero, 2004, Torero 2005), and the NEMI plug-and-play factory project (Dugenske 2000, CAMX 2001, Dugenske 2001) based on a communication-centric approach. Three major existing initiatives can be described within the scope of a device centric approach to factory automation, namely the ODVA architecture, the PROFINet architecture and the SIRENA architecture.

The ODVA architecture (CIP, 2008, CIP Network, 2008) defines a homogeneous platform for the development of process control applications. It is based on the CIP (Common Industrial Protocol) defined by ODVA. The CIP is an application level protocol based on the basic ODVA communication protocol stacks, namely DeviceNet, ControlNet, CompoNet, and Ethernet/IP, therefore covering all areas of the process

---

### *RI-MACS: An Innovative Approach for Future Automation Systems*

automation, from field level communications, up to production line control. The device-oriented approach of the CIP ensures encapsulation and modularity, enabling multi-vendor device solutions to co-exist on the same automation platform. ODVA also provides a safety object library along with device profiles that ensure reliable message exchange for safety-critical functionalities at CIP level. Although the ODVA architecture provides real-time support through the DeviceNet protocol stack, it does not support explicit QoS specification and management.

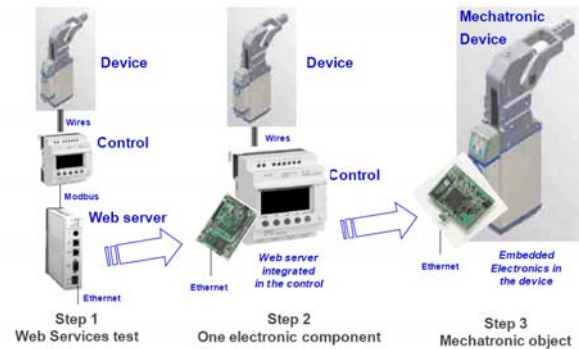
The PROFINet architecture (PROFINET, 2006) is similar to the ODVA architecture. It is a device-oriented approach based on the PROFIBUS at field level, COM and Ethernet at control application level. Similar to the ODVA initiative, the PROFINet architecture does not provide explicit QoS specification and management, although it does support real-time communications via the PROFIBUS protocol.

The SIRENA project (Jammes , 2005a, 2005b) advocates a significantly different approach, where service description and invocations are based on Web Service technology and a service oriented architecture. This approach has the significant advantage of exploiting a ubiquitous technology, which is intrinsically Internet compliant. Consequently, it offers the potential to enable factory integration more widely and openly than the preceding approaches. Moreover, the SIRENA project introduces the exploitation of the DPWS protocol for the plug-and-play integration of devices in the automation platform. A basic disadvantage of Web Service technology, as it is currently available, is the dramatic overhead introduced by the protocol, which limits its applications non-real-time applications.

The NEMI plug-and-play factory project (Dugenske, 2000,CAMX 2001, Dugenske, 2001) defines a framework for standard communications between electronic devices. This framework is based on the standard language XML and the resulting technology is called CAMX. Although CAMX favours interoperability among devices, it does not ensure interoperability at application level.

The TORERO (Torero 2004, Torero 2005) device-centric approach is based on Java technology to provide plug-and-play functionality. The concept of a device in the TORERO architecture is based on control application being distributed over multiple devices with no need for a central PLC to coordinate device operations. The overall control application can be developed using the TORERO IDE (Integrated Development Environment). This automation platform does not provide any Quality-of-Service management support.

**Figure 2:** Stepwise migration path for the integration of web services with control device functionality



### 3.2 Real-Time Requirements

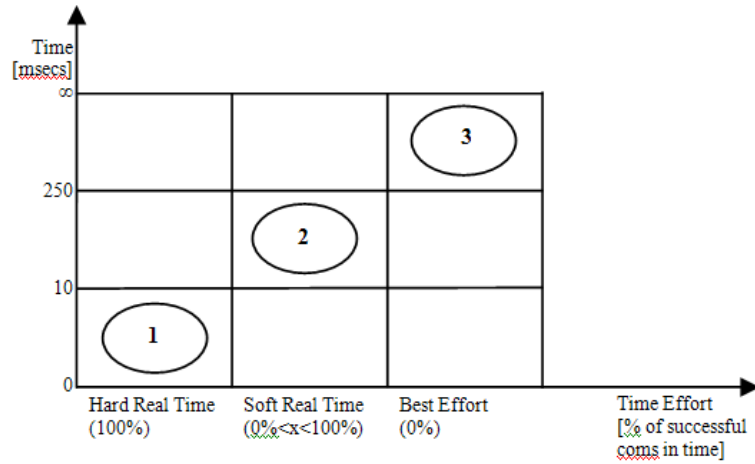
In typical process automation systems three basic classes of timing requirements can typically be identified: 1) control and I/O scanning, 2) HMI operations and interactions, and 3) collection of statistical data.

The control and I/O, e.g., for the coordination and interfacing, both locally and via networks, of valves, limit switches, clamps, rollers and other moving objects of automated machinery, require timing performance within 10 milliseconds. Their timing profile is subjected to these requirements to avoid collision between moving objects, and consequently requires a small reaction time. HMI operations and interactions involving slower user-oriented devices do not typically require such rapid response times and usually are subjected to timing requirements in the order of hundreds of milliseconds. The collection of statistical data from the devices is typically not subjected to strict timing constraints and response times in the order of seconds are usually acceptable.

Operations that fall in the first class above are usually critical and any missed deadline, i.e. response times above 10 milliseconds, are considered as faults that may cause possible failures in the system. Therefore, operations in class 1 are additionally required to be 100% satisfied (no deadline can be missed) and are considered in the RI-MACS project as “hard real-time” services. Deadline misses for operations in class 2 do not usually provoke system failures, so that they are considered “soft real-time” services, where a percentage of service invocations may be satisfied later than their nominal deadlines. Finally, operations in class 3 are considered to be subjected to best effort services, in that no formal timing deadline is required to be met, and the system is only required to do the best it can achieve in terms of response time. These requirement classes are illustrated in Figure 3. It is important to note that operations in classes 2 and 3 must typically still operate without any loss of data and with error notification if significant propagation delays have occurred.



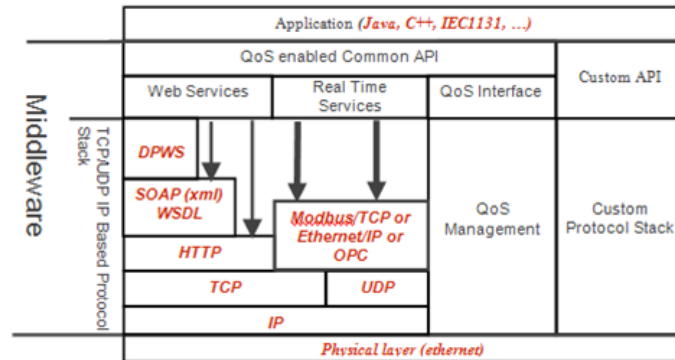
Figure 3: Process Automation Requirement Classes



### 3.3 The RI-MACS Architecture

In the RI-MACS architecture (see Figure 4), the application accesses the RI-MACS automation services through the RI-MACS API (Application Programming Interface). The service invocation can be directed towards the remote device through three different protocol stacks: the Web Service protocol stack, the real-time service protocol stack or the custom protocol stack.

Figure 4: The RI-MACS Service Architecture



The custom protocol stack is made accessible to the application through the custom API, which is not defined by the RI-MACS platform and is conceived with the sole purpose of

*RI-MACS: An Innovative Approach for Future Automation Systems*

allowing integration with legacy subsystems. It is required that this API be used only by peripheral modules of the control applications.

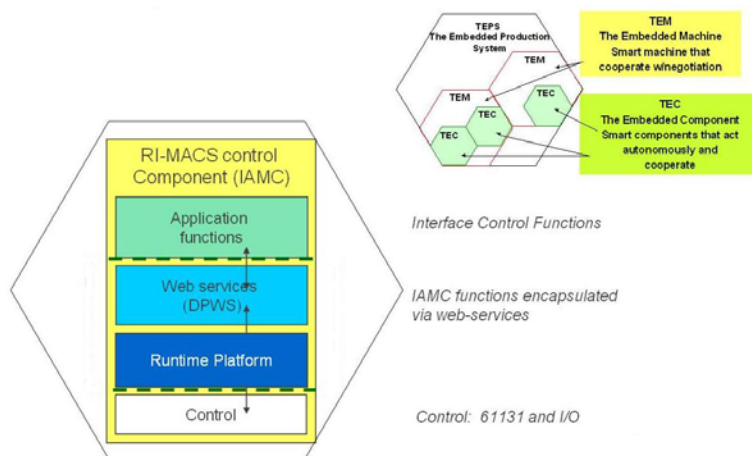
Due to the relevant overhead introduced by the SOAP (SOAP 2007) (Simple Object Access Protocol) messaging infrastructure, the Web Service protocol stack is currently limited to implement plug-and-play functionality, i.e. automatic device configuration, and invocations towards best effort and soft real-time services, such as diagnostic and maintenance services. It has been shown that today the Web Service protocol stack is not able to provide effectively a response time below tens of milliseconds (Jammes 2005a), although it is envisaged that in the near future the advancement of technology will address this limitation.

While Web technologies are gradually gaining ground in the embedded systems world, currently hard real-time services still require the implementation of a dedicated and efficient protocol stack. For this purpose, the RI-MACS architecture requires a dedicated real-time protocol stack, be implemented using real-time communication protocols, such as Modbus/TCP, Ethernet/IP, and ProfiNet.

Best effort, hard and soft real-time services are invoked through the same common API. Each service invoked from the common API is associated to a QoS specification. The QoS interface selects the correct protocol stack according to the QoS specification associated to the invoked service. The QoS management component of the RI-MACS architecture separates the different protocol stacks and the different QoS levels, in order to guarantee that non real-time services do not influence real-time service guarantees.

The component architecture of RI-MACS, which utilises these services, is summarised schematically in Figure 5. Embedded smart components at the device and machine levels must cooperate to provide the necessary overall system functionality.

**Figure 5:** The RI-MACS Component Architecture



In the RI-MACS architecture the automation hierarchy consisting of The Embedded Components (TEC), which form part of The Embedded Machines (TEM), which in turn

*RI-MACS: An Innovative Approach for Future Automation Systems*

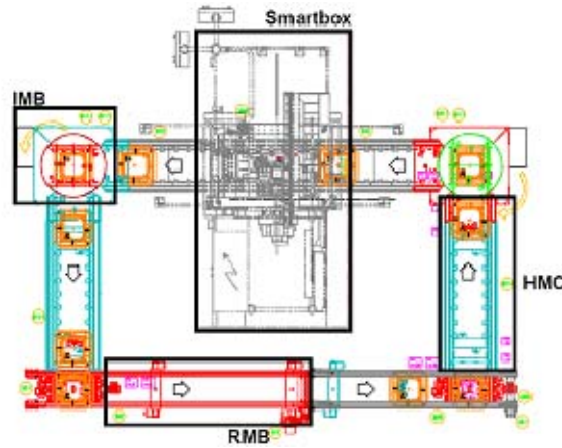
are part of The Embedded Production Systems (TEPS) (Bepperling 2006). In the project this architecture has been applied to a range of machines in the electronics and automotive assembly automation domains. An example application of a full-scale demonstrator test cell at COMAU is shown in Table 1.

**Table 1:** Module breakdown at COMAU

MODULE	EMBEDDED LEVEL
CONVEYORS (LEGACY)	TEC
TURNTABLES (LEGACY)	TEC
IMB	TEC
RMB	TEC
SMARTBOX	TEM
HMC	TEM
ENGINE ASSEMBLY CELL	TEP

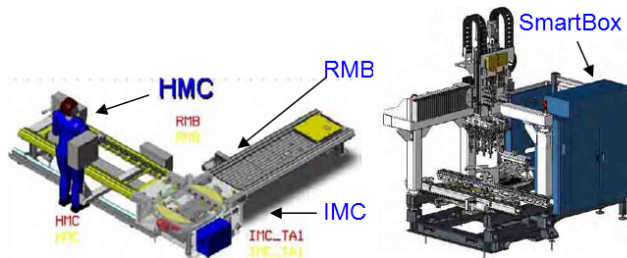
The architectural decomposition is shown in figure 6. This cell incorporate a combination of legacy devices with external controls interfaced via Web-servers and the higher levels with a set of Ethernet based mechatronic modules with embedded control functionality in the units themselves.

**Figure 6:** Module breakdown and layout of full-scale prototype assembly cell at COMAU



Namely the IMB rotary table, RMB linear conveyor, HMC manual station module and SmartBox flexible assembly station; see Figure 7.

**Figure 7:** Mechatronic modules within the COMAU assembly cell



---

### *RI-MACS: An Innovative Approach for Future Automation Systems*

Within RI-MACS architecture services are grouped in three main categories: general web services, automation oriented services and real time services, as illustrated in Table CC.

General web services are broadly similar to other web-oriented applications, typically based on DPWS, providing an efficient way to discover other DPWS enabled devices, their name, and capability and communication requirements.

Automation oriented web services, are a group intended to cover the needs of the mechatronic device communication. These services enable remote device access and simplify the code that is executed in the centralized controller by encapsulating additional functionality within devices that is traditionally too expensive to code inside a centralized architecture.

Configuration services, provide a new simple, standardised way to modify and store the parameters of a device, this operation is traditionally done through custom serial cables or via localized custom operator interfaces. For example, consider a frequency inverter; this is generally sold with an ad-hoc small numeric keypad that enables the configuration of parameters along with a four-digit display. To simplify such operations, in the RI-MACS platform a standard web page is provided that enables access to such devices over the Ethernet from any Web browser.

The diagnostic interface provides details of problems raised by the device. The interface also makes it possible to move part of the diagnostic functionality from the centralized controller, where the diagnostic code is generally written in a custom way to the local device, where it can be standardized, thus in this way is possible to simplify the coding.

The trace service provides a convenient way to collect statistical data on the device that can be used to populate a database of the devices efficiency. This data is collected in the background when the device is working, and is transmitted on the network with a low priority. In this way, the operation does not affect the performance of the machine. The data to be collected can be of various types, such as:

- the number of a fault in order to understand if the device has to be changed or fixed;
- the number a of task performed in order to evaluate the remaining life of the device;
- the time to perform a task, in order to understand the deterioration of the performance as a function of the number of operations performed.

The maintenance services can provide specific data on the device, ranging from schematics to the lifetime of the device in order to replace it in time.

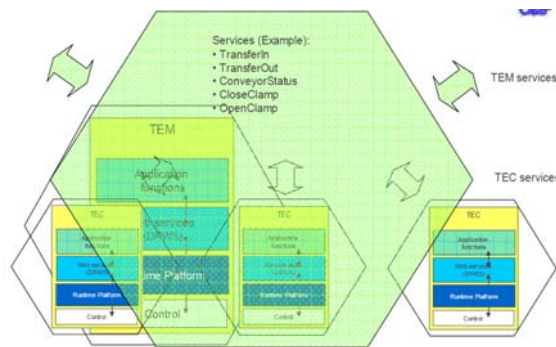
The control interface allows an operator to access the functionality of the device through a web page, in this way the execution of the operator interface can be moved from the centralized controller to the local device.

The hard real-time services provide privileged access to the encapsulated functionality of the device, guaranteeing cycle-time performance and avoiding collisions with other devices.

### 3.4 .1 Example Services

As illustrated in Figure 8 the concept which has been applied in RI-MACS is the availability of appropriate Web-service functionality at the TEC, TEM and TEP levels coupled with support for real-time (non Web based) services where necessary, e.g., where performance or safety issues make this necessary.

**Figure 8:** Concept of a hierarchy of Services supporting the required functionality



As an example, Table 2 shows the service interface of the IMB mechatronic module. A similar approach has been adopted for all modules. The services are categorised into three groups, general Web services, automation related Web-services and real-time, non Web based, services. The time profile relates to the quality of service required and real-time behaviour, be it a wired or wireless interface which is used – see section 3.2 and 4.3 for further details. Column three in the table indicates the type of service functionality, e.g., DPWS indicates a general service, automation specific functionality has been subdivided into configuration, diagnostics, behavioural tracing, maintenance support, and basic control functions. Not the services described so far have non/soft real-time requirements which can be accommodated within the capabilities of current Web-service technologies. For example direct control is limited to the ability to stop and start the automatic cycle of the module and it not related to its movement behaviour interlocked with other machine modules. Such synchronisation is achieved within the third group of real-time, profile 1, services which are not Web-based.

**Table 2:** Example TEC module service interface for IMB rotary table

General Web Services	Time behaviour profile: 2	DPWS	Eventing
			Discovery
			.....
Automation oriented web services	Time behaviour profile: 3	Configuration	GetConf
			setConf
	Time behaviour profile: 2	Diagnostic	startDiag
			setDiag
			getDiag
	Time behaviour profile: 3	Trace	startLog
			stopLog
			getLog
	Time behaviour profile: 3	Maintenance	getMaintInfo
	Time behaviour profile: 2	Control	Stop End Of Cycle
Start automatic			
Hard Real Time	Time behaviour profile: 1	Synchronization	Modbus/TCP
			Ethernet/IP
			Device access services

### 3.5 Quality of Service and its Management

The central characteristic of QoS in the RI-MACS automation platform is that it is both flexible and can also support the required service characteristics in various levels of automation architecture. Within RI-MACS QoS can be measured at three main levels. These can be viewed in a hierarchy from top down and consist of Business level QoS, Application level QoS and Device level QoS.

Business level QoS is an area of research that has developed alongside SOA development and use. Business level QoS is linked to the provision of Service Level Agreements (IBM 2003) between service users and providers. For example a factory producing certain goods will guarantee in a automated e-business system that it can match certain production metrics for its customers. The SLA (Service Level Agreement) could state this together with any penalties to be enforced if the production target is not met. These business / higher level agreements link low level Quality-of-Service metrics to business level SLA. These the SLAs are intrinsically linked to QoS metrics and directly reflect the manufacturing environment output.

At the Application level, the management of QoS is structured in three layers: the service layer, the architectural layer and the resource layer. In the service layer, the QoS specification is provided in terms of timing requirements. This QoS specification is associated with the manufacturing service (machine / line) characteristics. In the architectural layer, the QoS management is represented in middleware services. These services are responsible for linking the top and bottom layer. Services exist to aid the establishment of such links, and include services that provide functions such as QoS level negotiation, resource reservation, QoS level monitoring and QoS exception management. The expression of low level QoS service requirements are translated into resource reservation parameters. To provide effectively QoS guarantees, the underlying

implementation platform must provide support for the implementation of resource reservation techniques.

The final level of QoS can be seen in the automation hierarchy as the device level QoS. A device can be seen as an element that makes up a service such as a conveyor. These can include the sensors, actuators and networking equipment. QoS metrics are sourced at this level and are measured in finite terms such as response time. These metrics of real-time performance are vital in any production system and as already discussed in terms of response-time are important issues for successful adoption of a SOA approach.

The bridging of QoS metrics into SLA is vital in the RI-MACHS applications. The three layers introduced allow the control and monitoring of the lowest level devices to be included in the higher-level business models. This QoS metric management also enables the quick reconfiguration of lines by presenting clear channels of performance and measurement metrics in wider business applications.

## **4 RI-MACS Integrated Design Environment**

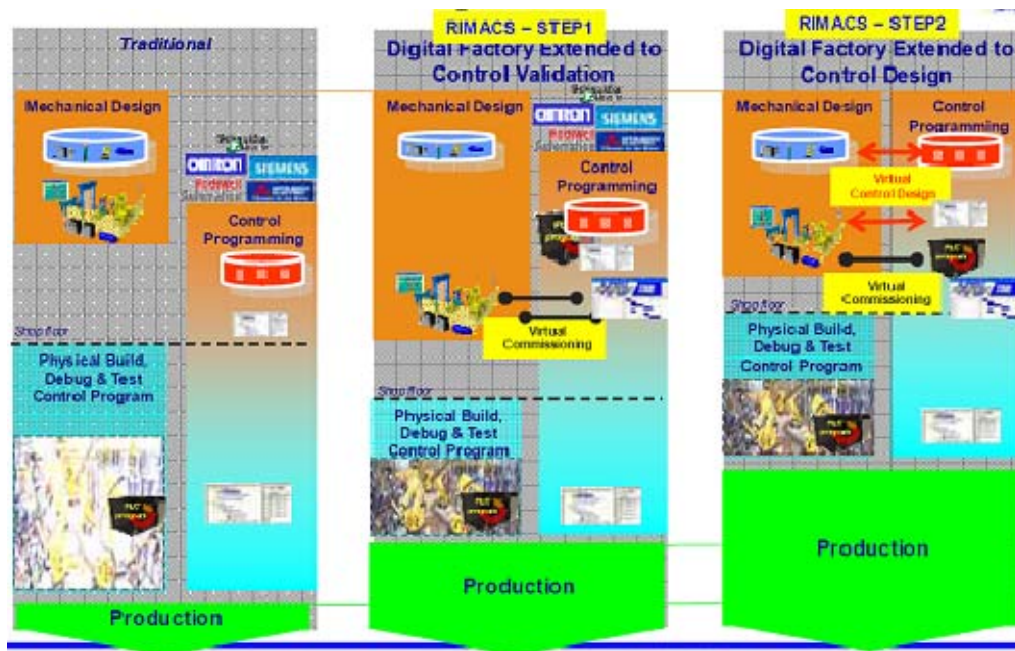
### **4.1 Introduction to Engineering Methods**

In the domain of control system development, the issue of control code testing is very critical: this is particularly true when considering the complex aggregation of many control functions.

Currently, it is common in industry to subcontract and reuse control solutions. Assigning some of the control code development to external software development companies is becoming more common. Nevertheless, it highlighted the problem of how to validate effectively incoming solutions (e.g., from sub-contractors) against the original specifications and requirements. Reusing control solutions that have already been developed for previous applications is also very desirable. However, the correctness of available existing solution with respect to a particular new application case is typically very difficult to currently verify.

The complexity derived from the interactions of various portions of control code call for new innovative approaches that can enable the verification and the validation of the developed solution, and the provision of an appropriate means to allow proper interpretation of the obtained result. The detection of errors and bugs in the developed control code has the potential to significantly shorten the commissioning phase of automation projects as illustrated in Figure 9. Beyond virtual commissioning, however there is potential to achieve more complete, verified control system design and lifecycle support as illustrated in step 3 of Figure 9. The new engineering capabilities that have been implemented in RI-MACS have taken a major step in the provision of such capabilities.

Figure 9: Application of enhanced digital engineering capabilities with RI-MACS



## 4.2 New Engineering Capabilities

Two key aspects of engineering tool capabilities have been targeted within the RI-MACS research: first, a method to enable machine nominal behaviour to be input in a manner with which machine builders are familiar, i.e., cycle timing diagrams; and second, an integrated method to enable machine modules to be incrementally tested and verified against the required behavioural specification. These capabilities have been implemented as enhancements to the Delmia Automation V5 digital engineering toolset. However, the interfaces are predominantly XML-based, and so integration with other toolsets

Within the RI-MACS project, the approach followed in the establishment of a design methodology and tools to support the development of control system is structured as follows:

- A specific RI-MACS toolset has been developed, allowing the definition of the control code nominal behaviour in a Gantt chart formalism.
- A transformation of the Gantt chart is carried out into an executable formalism, namely SFC, in order to perform the export into the DELMIA Automation V5 environment. This step will therefore result in the creation of the nominal control code portion, or Control Logic (CL), according to DELMIA terminology.

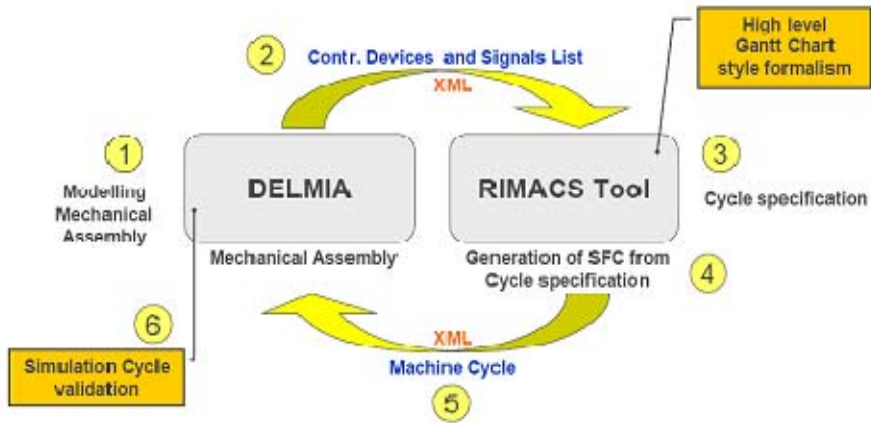


*RI-MACS: An Innovative Approach for Future Automation Systems*

- The integration of the CL with the 3D simulation aspects of the system are performed in DELMIA V5: this implies the management of 3D graphic models, resulting from the import of CATIA (or other CAD) files of the controlled objects, animated according to specific behaviour – the so-called Internal Logic.

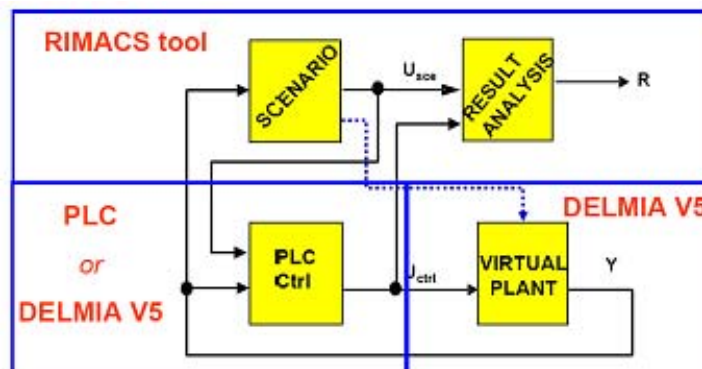
The procedure for use of the RI-MACS cycle-timing specification tool, illustrated in Figure 10, is as follows. The mechanical assembly should be modelled in Delmia Automation (1). As part of this procedure the list of control devices and their input-output signals are specified and then exported in XML format (2). Within the RI-MACS tool the behaviour and interaction between these devices can be specified (3) and the control logic generated from this machine cycle specification (4). The resultant control logic is then exported again via XML back to Delmia (5), and the machine cycle is then validated via simulation (6).

**Figure 10:** Integration of new engineering functionality with Delmia Automation



The second significant extension to engineering tool functionality achieved within RI-MACS has been the ability to incrementally test and verify machine modules against their required behavioural specification. As illustrated in Figure 11, this approach involves integrating the PLC-based control code with a model of the controlled machine and then performing tests based on expected scenarios of operation for the system. There is within these scenarios the need to force fault conditions and analyse the machine behaviour in response to the various operational cases.

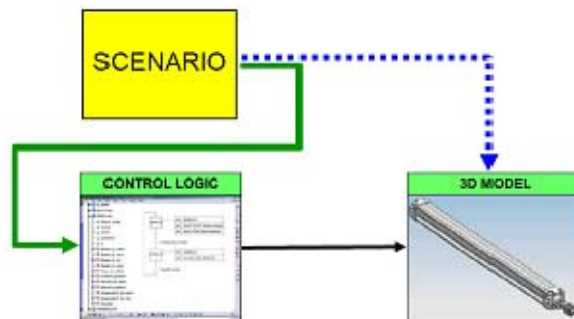
**Figure 11:** Testing and verification of machine modules against behavioural specification



Within the RI-MACS project, the interaction between the RI-MACS tool and the DELMIA V5 automation suite fosters a scenario in which the verification process can benefit from the use of a completely virtual model of the controlled objects. In order to achieve such benefits as effectively as possible, a simulation based verification technique has been defined. The execution of the control logic algorithm is compared to a boolean temporally dependant sequence. Thus, by triggering boolean values and simulating temporal delays, branches in the control code can be executed. The problem of verifying the correctness of such an algorithm is an NP (Nondeterministic Polynomial) time problem, so the verification performed via simulation cannot be exhaustive, due to intrinsic complexity of the problem. Nevertheless, an application-oriented approach, in which proper testing is conducted according to some patterns of validation, can help discovering a significant quantity of errors present in the control code, so contributing to achieving a robust control application, and reducing the time to market in the commissioning phase, as previously mentioned.

According to such a technique, whenever a single module of the automation system has been designed, the designer shall follow and consequently build up a testing scheme to verify the compliance of the designed module with respect to its specification, see Figure 12.

**Figure 12:** Concept of incrementally testing and verifying the behaviour of individual automation components



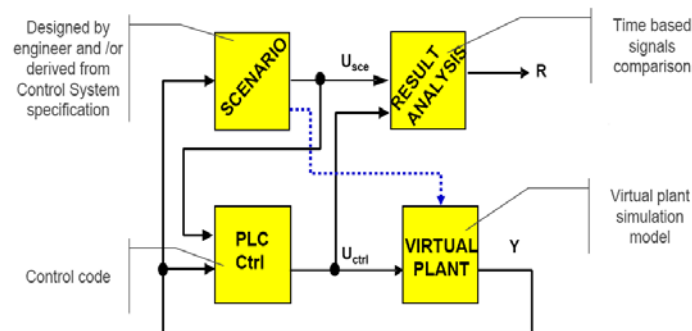
Such a scheme is defined by properly parameterizing simple templates, that represent the closed-loop system, where the process to be controlled is suitably simplified. An automatic procedure can then be launched to test the designed module. Such a framework can then be adopted to define a structured bottom-up methodology for testing the overall automation system.

In detail as shown in Figure 13:

- PLC Controller – contains the control functions to be tested;
- Virtual Plant -- represents a simplified model of the controlled sub-system;

- Results Analysis -- contains a formal description of the desired behaviour, whose definition is up to the designer, and which compares the simulation results with such a desired behaviour; and
- Scenario – contains the input signal sequences that should be generated to test the PLC Controller via simulation.

**Figure 13:** Simulation technique schema



The Virtual Plant block describes the system portion controlled by the considered PLC; usually, such a system portion is constituted by both process and control components. In order to test the PLC it is not necessary, in principle, to define an accurate model for the controlled object. Often, a simple model that represents the feedback reaction to the PLC commands is adequate for the purpose. The defined scheme can be used to automatically check that, for a given input value or sequence, the PLC computes the outputs according to the desired specification. An exhaustive testing of a PLC is often unfeasible for a significant portion of a factory automation system, because of the excessive number of possible input combinations. Consequently a significant problem is the definition of the test cases, i.e. of the input signal (I in figure 5) sequences that should be generated to test the PLC via simulation.

An important role is played by the Result Analysis module, which must consider the behaviour of the Scenario (i.e. the desired behaviour from the executable specification) and compare this with the behaviour of the PLC (i.e. real control code behaviour) perform a comparison between the two, and highlight any discrepancies.

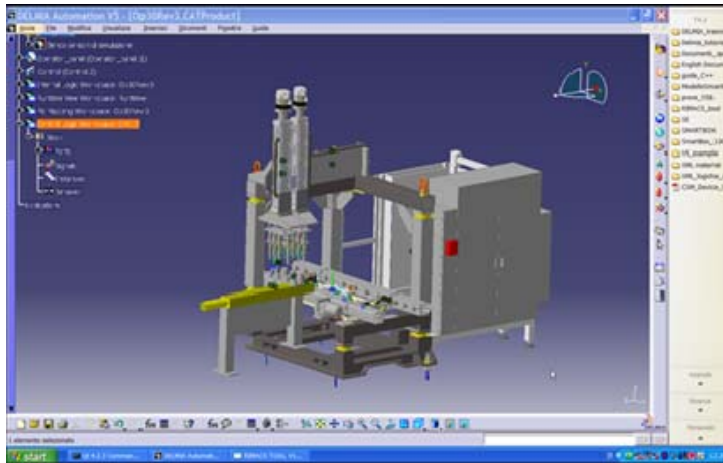
Once a PLC control code has been designed, in order to apply the proposed testing method, the following problems have to be faced:

- modelling the Virtual Plant;
- modelling of the desired behaviour and definition of the Results Analysis module;
- generation of suitable Scenarios of input signal sequences.

## *RI-MACS: An Innovative Approach for Future Automation Systems*

Since an automation system is usually structured according to different hierarchic levels, the testing of the overall automation system from the bottom-up approach is a valid and practical approach. Each hierarchical level is typically composed of more PLC code modules, and such modules can be tested “one level at a time”. First, code modules at the lowest level with respect to the functional hierarchy are tested according to the concept shown in figure 4 and 5. Therefore, the required Virtual Plant has to be defined to model the field responses. Progressively higher level PLC code modules can be tested assuming that modules at the lower levels have been already tested. In such a case, the Virtual Plant for testing the current level code modules can be realised directly through the already tested lower level PLC code modules and the Virtual Plant representing the field of the lower levels. Following the above ideas approach a complete bottom-up testing procedure for a control system can be practically realized. Figure 14 illustrates a sample application tool used within the RI-MACS automation system.

**Figure 14:** Typical model of a RI-MACS automation system in Delmia Automation



## **5 Conclusions**

This paper has presented the overall vision of the RI-MACS research project. It has highlighted key industrial needs for innovative solutions to automation. Three key aspects of the research have been described: the RI-MACS automation architecture, the integrated design environment, and the need to address quality of service issues in order that distributed automation systems can provide the required functionality and performance. The RI-MACS control architecture has achieved the integration of Web-service capabilities with traditional real-time PLC functionality. The architecture also categorizes services in terms of their performance requirements. This is a first step

---

### *RI-MACS: An Innovative Approach for Future Automation Systems*

towards a more comprehensive approach to quality of service management within automation systems. The RI-MACS design tools have extended the capabilities of digital automation in terms of specification of machine nominal behaviour via timing diagrams and the creation of a methodology to enable the incremental testing and verification of machine modules via a bottom-up approach leading to complete system verification and facilitating reuse. On-going work is now realising the RI-MACS approach in prototype form at the industrial partner sites.

The RI-MACS approach is advancing the automation control systems by reducing costs and complexity. From a business and in particular a customer focused perspective, RI-MACS aims to meet the ever decreasing time-to-market constraints, while at the same time increasing the capability to adjust quickly within production plants to the rapidly evolving market scenarios. To achieve this result, the concept of encapsulated mechatronic devices is central to the RI-MACS philosophy. The introduction and use of Web technology in service oriented architectures promises to reduce drastically the amount of custom software in the main controllers of the production line, that then only need to focus on the synchronization of the sequence of the production operations. Moreover, the future development of QoS specifications and management support will be a key enabler for the scaling up of the RI-MACS approach to a full-fledged dynamic automation platform capable of predictably addressing large applications of significant complexity.

### **ACKNOWLEDGMENT**

The authors gratefully acknowledge the support for this work from the RI-MACS (Radically Innovative Mechatronics and Advanced Control Systems – Ref. NMP2-CT-2005-016938) European Framework Programme Six project.

### **REFERENCES**

- Bepperling A., Mendes J., Colombo A. W., Schoop R., Aspragathos A. (2006) “A Framework for Development and Implementation of Web service-Based Intelligent Autonomous Mechatronics Components.” INDIN 2006
- Brevi D., Mazzocchi D., Scopigno R., Bonivento A., Calcagno R., Rusina F., (2006) “A Methodology for the Analysis of 802.11a Links in Industrial Environments”, IEEE International Workshop on Factory Communication Systems, Page(s): 165 – 174
- Calcagno R., Bonivento A., (2006) Wireless teach pendant for robotics technological rationale for Comau WiTP”, Eighth International IFAC Symposium on Robot Control, Volumn 8, Part 1, SYROCO
- CAMX (2001) “Generic Requirements for Electronics Manufacturing Shop-Floor Equipment Communication (CAMX)”, IPC-2541
- CIP (2008) “Common Industrial Protocol (CIP)”, [http://www.odva.org/Portals/0/Library / Publications\\_Numbered /PUB00122R0\\_CIP\\_Brochure\\_ENGLISH.pdf](http://www.odva.org/Portals/0/Library/Publications_Numbered/PUB00122R0_CIP_Brochure_ENGLISH.pdf)
- CIP Network (2008) “The CIP Networks Library”, <http://www.odva.org>
- Dugenske A., “The Framework Implementation Project”, Circuits Assembly, March 2001, [http://thor.inemi.org/webdownload/newsroom/Articles/CA-dugenske-3\\_01.pdf](http://thor.inemi.org/webdownload/newsroom/Articles/CA-dugenske-3_01.pdf)
- Dugenske A., Fraser A., Nguyen T., Voitus R., “The National Electronics Manufacturing Initiative (NEMI) plug and play factory project”, International Journal of Computer Integrated Manufacturing, 2000, Vol. 13, No. 3, pp. 225-244, <http://thor.inemi.org/>

---

*RI-MACS: An Innovative Approach for Future Automation Systems*

- webdownload/newsroom/ Articles/IJCIM0500.pdf
- IBM (2003) “Web Service Level Agreement Language Specification, Version 1.0”, IBM Corp., <http://www.research.ibm.com/wsla/>
- IBM (2005), “Service-Oriented Architecture”, Special issue in IBM Systems Journal, Volume 44, Number 4, 2005
- Jammes F., Smit H. (2005b) “Service-Oriented Paradigms in Industrial Automation”, IASTED`05, International Multi-Conference, Parallel and Distributed Computing and Networks
- Jammes F., Smit H., (2005a) “Service-Oriented Architecture for Devices – the SIRENA View”, INDIN05, August 2005, <http://www.sirena-itea.org/Sirena/Documents/Publications/SortedDocumentsList.htm>
- OASIS (2006) “OASIS Reference Model For Service Oriented Architecture V 1.0”, Official committee specification, 2006, <http://www.oasis-open.org/committees/download.php/19679/soa-rm-cs.pdf>
- PROFINET (2006) “PROFINET Technology and Application – System Description”, [http://www.profibus.com/celummdb/doc/PROFIBUS/Downloads/Technical\\_Description/4132\\_system\\_description\\_april2006.pdf](http://www.profibus.com/celummdb/doc/PROFIBUS/Downloads/Technical_Description/4132_system_description_april2006.pdf)
- SOAP (2007) “SOAP Version 1.2 Part 0: Primer”, Apr. 27, <http://www.w3.org/TR/soap12-par>
- Torero (2004) “Total life cycle web-integrated control - Overview”, [http://research.altec.gr/torero/html/documents/torero\\_flyer.pdf](http://research.altec.gr/torero/html/documents/torero_flyer.pdf)
- Torero (2005) “TORERO: Total life cycle web-integrated control”, [http://research.altec.gr/torero/html/documents/torero\\_panel.pdf](http://research.altec.gr/torero/html/documents/torero_panel.pdf)