

EUROPEAN COMMISSION

**Thematic Priority:
SIXTH FRAMEWORK PROGRAMME**



**Priority 2.5.3
INFORMATION SOCIETY TECHNOLOGIES
Unit G3 Embedded Systems**



Project Acronym:

SOCRADES

Project Full Title:

**Service-Oriented Cross-layer infRAstructure for
Distributed smart Embedded devices**

Proposal/Contract No: EU FP6 IST-5-034116 IP SOCRADES

Deliverable D7.2

Investigation of Enabling Technologies, Methods and Tools for Application Engineering

Status:	Final
Dissemination Level:	CONFIDENTIAL
Date:	01.10.2007

Organization Name of the Lead Contractor for this Deliverable: **Loughborough University**

Status description:

Deliverable number:	D 7.2		
Deliverable name:	Investigation of Enabling Technologies, Methods and Tools for Application Engineering		
Version:	Final	Resources consumed:	[13.9 in total] pm
Scheduled completion date:	[31.08.2007]	Actual completion date:	[24.09.2007]
Author(s)/ deliverable:	R.Harrison (Lboro), R.Monfared (Lboro), D. Vera(Lboro), F. Depeisse (Schneider) A. Bepperling (Schneider)	Report/deliverable classification: <input checked="" type="checkbox"/> Deliverable <input type="checkbox"/> Partner report <input type="checkbox"/> Partner report (aggregated) <input type="checkbox"/> Task report <input type="checkbox"/> Work package report <input type="checkbox"/> Three-Monthly Activity Report <input type="checkbox"/> Six-Monthly Activity Report	
<input type="checkbox"/> <input type="checkbox"/> Partner ↓ ↓ Peer reviews ↓ ↓ Contributions	<input checked="" type="checkbox"/> <input type="checkbox"/> Schneider Electric <input type="checkbox"/> <input type="checkbox"/> ABB <input checked="" type="checkbox"/> <input type="checkbox"/> APS GmbH <input type="checkbox"/> <input type="checkbox"/> Boliden AB <input type="checkbox"/> <input type="checkbox"/> FlexLink Automation Oy. <input type="checkbox"/> <input type="checkbox"/> Institut f. Automation und Kommunikation e.V. Magdeburg <input type="checkbox"/> <input type="checkbox"/> Kungliga Tekniska Högskolan	<input checked="" type="checkbox"/> <input type="checkbox"/> Loughborough University <input type="checkbox"/> <input type="checkbox"/> Luleå University of Technology <input checked="" type="checkbox"/> <input type="checkbox"/> Politecnico di Milano <input checked="" type="checkbox"/> <input type="checkbox"/> SAP AG <input type="checkbox"/> <input type="checkbox"/> Siemens AG <input type="checkbox"/> <input type="checkbox"/> Tampere University of Technology <input type="checkbox"/> <input type="checkbox"/> Jaguar Cars Ltd. <input type="checkbox"/> <input type="checkbox"/> ARM Ltd.	
Short description:	Investigation of enabling technologies, methods and tools		
Peer Review Approval:	<input checked="" type="checkbox"/> Approved <input type="checkbox"/> Rejected (improve as specified above)	Date:	26.09.2007
Presentation Comments:	Revision of section 11 (SAP). Check the figures.	Date:	27.09.2007

Table of Contents:

EXECUTIVE SUMMARY	5
1. INTRODUCTION TO THE ENABLING TECHNOLOGY WITHIN THE APPLICATION SYSTEMS ENGINEERING DOMAIN	6
2. EVALUATING THE ENABLING TECHNOLOGIES, TOOLS AND METHODS.....	7
2.1. CATEGORISATION AND FORMAT	7
2.2. SCOPE	8
3. SELECTED ENABLING TECHNOLOGIES	8
4. V-SIM CONTROLS VALIDATION SUITS.....	8
4.1. GENERAL DESCRIPTION	8
4.2. OVERVIEW OF CAPABILITIES	9
4.3. SUMMARY OF STRENGTHS AND WEAKNESSES	9
4.4. EVALUATION SUMMARY	10
5. DELMIA AUTOMATION.....	10
5.1. GENERAL DESCRIPTION	10
5.2. OVERVIEW OF CAPABILITIES	10
5.3. SUMMARY OF STRENGTHS AND WEAKNESSES	11
5.4. EVALUATION SUMMARY	12
6. UNIGRAPHICS EM TOOL SUITE	12
6.1. GENERAL DESCRIPTION	12
6.2. OVERVIEW OF CAPABILITIES	12
6.3. SUMMARY OF STRENGTHS AND WEAKNESSES	13
6.4. EVALUATION SUMMARY	13
7. PROCESS DEFINITION ENVIRONMENT - PDE.....	14
7.1. GENERAL DESCRIPTION	14
7.1. OVERVIEW OF CAPABILITIES	14
7.2. SUMMARY OF STRENGTHS AND WEAKNESSES	15
7.3. EVALUATION SUMMARY	16
8. CONTROLBUILD APPLICATION GENERATOR.....	16
8.1. GENERAL DESCRIPTION	16
8.2. OVERVIEW OF CAPABILITIES	16
8.3. SUMMARY OF STRENGTHS AND WEAKNESSES	16
8.4. EVALUATION SUMMARY	16
9. UNITY APPLICATION GENERATOR (UAG)	17
9.1. GENERAL DESCRIPTION	17
9.2. OVERVIEW OF CAPABILITIES	17
9.3. SUMMARY OF STRENGTHS AND WEAKNESSES	17
9.4. EVALUATION SUMMARY	18
10. UNITY PRO.....	18
10.1. GENERAL DESCRIPTION	18
10.2. OVERVIEW OF CAPABILITIES.....	18
10.3. SUMMARY OF STRENGTHS AND WEAKNESSES	19
10.4. EVALUATION SUMMARY	19
11. SAP.....	19
11.1. GENERAL DESCRIPTION	19
11.2. OVERVIEW OF CAPABILITIES.....	19
11.3. SUMMARY OF STRENGTHS AND WEAKNESSES	20
11.4. EVALUATION SUMMARY	20

12. CATALOGUE OF INVESTIGATED TECHNOLOGIES	20
13. EVALUATION OF REVIEWED ENGINEERING TOOLS	22
14. SERVICE-BASED AND COMPONENT-BASED SYSTEMS ENGINEERING	23
14.1. TORERO.....	24
14.2. OCEAN.....	24
14.3. SIRENA	25
14.4. ABAS	25
15. OVERALL CONCLUSIONS	26
16. REFERENCES	26

List of Figures:

Figure 1: Dimensions of collaborative integration and support.....	6
Figure 2: V-Sim User Interface	9
Figure 3: Delmia Automation Overview (source Delmia)	11
Figure 4: COMPAG PDE Overview	15
Figure 5: Description of a TORERO Device	24
Figure 6: OCENA Infrastructure overview	25
Figure 7: Overview of SIRENA Framework	25
Figure 8: Coverage of Virtual engineering enabled technologies	21
Figure 9: Coverage of Control specific technologies.....	22

List of Tables:

Table 1: List of investigated enabling technologies and tools	8
Table 2 : Evaluation of V-Sim based on 4-dimensional criteria.....	10
Table 3 : Evaluation of Delmia Automation based on 4-dimensional criteria.....	12
Table 3 : Evaluation of Unigraphics eM tool suite Automation based on 4-dimensional criteria	13
Table 4 : Evaluation of PDE based on 4-dimensional criteria	16
Table 4 : Evaluation of ControlBuild based on 4-dimensional criteria.....	16
Table 5 : Evaluation of UAG based on 4-dimensional criteria.....	18
Table 6 : Evaluation of UnityPro based on 4-dimensional criteria.....	19
Table 8 : Evaluation of SAP based on 4-dimensional criteria	20
Table 9: Summary of evaluated Enabling Methods and Tools.....	20

Executive Summary

This study is focused on a comparison of practically available enabling technologies for application systems engineering, which might potentially be utilized within the scope of the SOCRADES project. The coverage is not broad and shallow but rather takes a selective and critical look at available candidate engineering tools. A discussion is also included of component-based system representation and modelling in the manufacturing domain.

Summaries of strengths and weaknesses and the evaluations of the engineering tools presented provide a broad indication of the capabilities of these tools against selected criteria for control, enterprise integration, supply-chain and lifecycle support, and virtual engineering. In the work to create appropriate engineering tools for the SOCRADES project, this information will be helpful in terms of understanding what aspects of current toolsets are useful in the context of the applications being studied. An important finding has been that most current engineering systems are considered too complex and general purpose and that users typically want an engineering system focused on their specific needs. In this respect, the adoption of a component-based approach offers the opportunity to migrate towards engineering methods based predominantly around the configuration of application-specific components rather than the adaptation of general-purpose controllers such as PLCs.

The set of tools reviewed in this document can be classified into two main categories depending on whether they are oriented towards a) virtual engineering or b) control software engineering.

Providing a visualization environment to assist the end-user and enabling visualizations to be linked to the physical control components are important engineering requirements needed by end-users. Software tools sometimes referred to as “virtual engineering” introduce a set of functionalities related to the design and implementation of a 3D-based virtual model. This set of functions aims at providing intuitive views of the system being designed, the main objective being to break down the barriers that typically exist among control, process and mechanical engineers. The second challenge is to achieve effective integration between 3D modelling and control data so that a model’s behaviour can be consistently maintained as a combination of control logic and dynamic model animation in order to conduct process and sequence analysis.

Engineering environments oriented towards control editing and hardware control code generation aim at assisting control and process engineers to manage the task of implementing control code for complex/large-scale systems by a) providing direct translation of process production sequence data into usable code at hardware level and b) providing means to maximise re-use and re-configuration of control code.

No single available engineering tool reviewed here adequately supports control and enterprise and supply chain integration. The paradigm of service-oriented systems provides the opportunity to create systems naturally from loosely coupled distributed components supporting control, enterprise and supply chain integration. The creation of suitable configuration and support tools for such systems is heavily dependent on the underlying system architecture. What can be supported within the engineering environment is a suitable approach for the specification, configuration, and behaviour of such components and a method of storing these for reuse. None of the tools looked at currently support Web services in the control domain, although ControlBuild, Delmia Automation, and PDE offer varying degrees of support for component-based systems and this model can be readily mapped to Web services.

1. Introduction to the enabling technology within the application systems engineering domain

The focus of WP7 is to support end-user needs for application engineering as efficiently as possible in the context of future SOA-based embedded systems. This deliverable investigates the capabilities of available enabling technologies, methods and tools for application systems engineering. The underlying service-oriented infrastructure and its management are covered elsewhere; see in particular WP5.

The aim here is to consider the control system in the context of the user's total business needs. As illustrated in Figure 1 the approach adopted throughout WP7 is to study automation systems from four key perspectives, namely in the control, enterprise integration, and supply chain and lifecycle contexts. D7.1 has considered user needs from these perspectives and D7.2 will now build on this by investigating the capabilities of current enabling tools and methods from the same four perspectives.

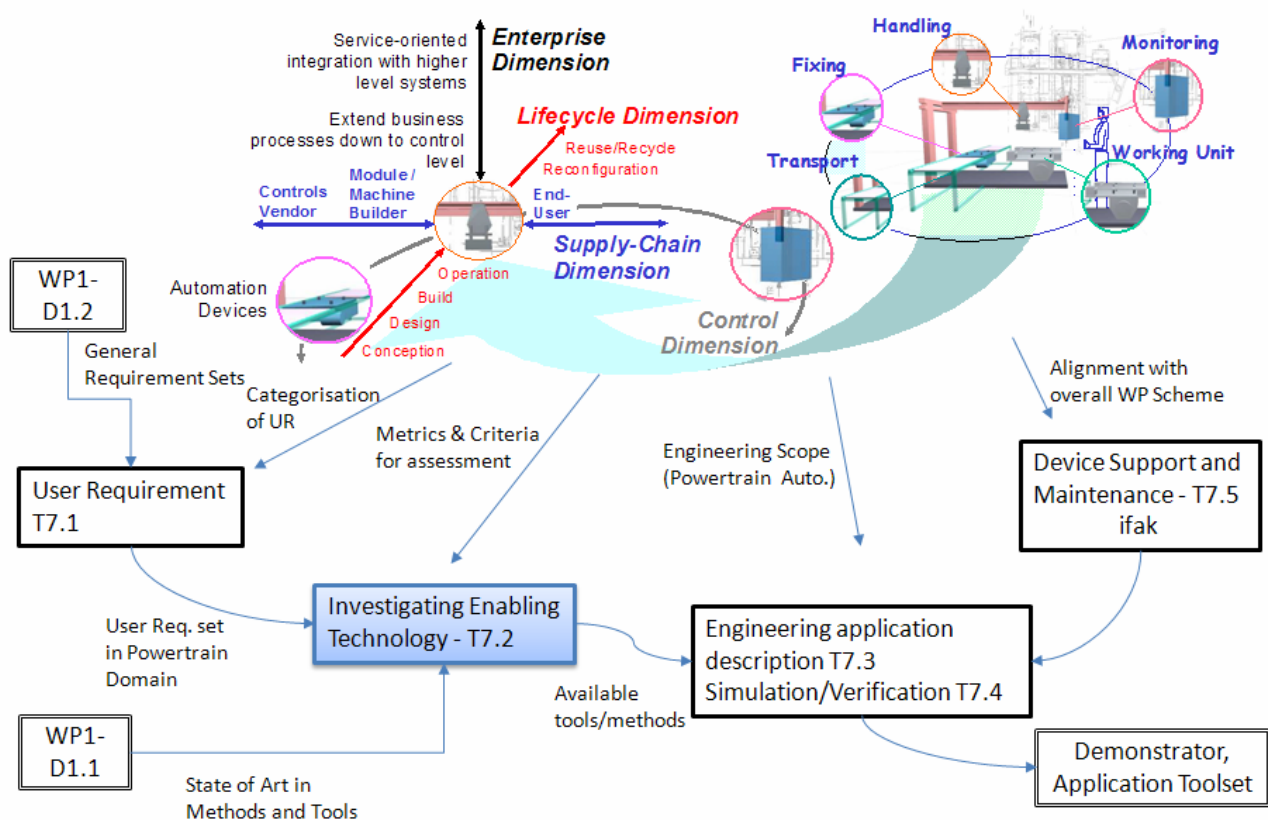


Figure 1: Dimensions of collaborative integration and support

As illustrated in Figure 1, D1.2 has already addressed the broad capture of basic user requirements across the SOCRADES project. From an application engineering perspective D7.1 has built on this by taking an in-depth look at the complete lifecycle and supply chains involved in selected application domains in which the prototype engineering tools to be created will be evaluated later in the project. For example, direct evaluation of the engineering approach in the automotive powertrain sector is planned with Jaguar. D7.1 has begun to identify engineering tool usage needs across the supply chain and over the complete lifecycle. Key user requirements established in D1.2 and D7.1 are to be used as the basis for D7.2, which seeks to assess the capabilities of current enabling technologies, methods and tools in the context of both today's needs and anticipated future needs for engineering configurable automation systems composed of distributed, smart embedded devices within the SOCRADES paradigm.

The outputs of D7.2 will be a direct enabler of the planned work in T7.3, T7.4 and T7.5 on engineering application description, simulation and verification, and device support and maintenance. The object is to

establish the capabilities of candidate tools and methods – both in the commercial and the research domains. In the wider project context the outputs of this work package also interrelate to WP5 where the service-centric infrastructure is being defined, to WP6 where the enterprise dimension of the project is being developed and ultimately to WP8 the application pilots.

This study is focused on a comparison of practically available enabling technologies, which might potentially be utilized within the scope of the SOCRADES project. The coverage is not broad and shallow but rather takes a selective and critical look at available candidate engineering toolsets.

From a control perspective, deliverable D1.1 has previously provided an overview of the engineering of distributed systems including control level application engineering methods, including IEC 61131-3, IEC 61499, Natural Languages, Flowcharts methods (e.g. Flow-Pro), template-based design methods (such as Error Dynamic Diagnostic Indicator system) and service-based application design. Business applications engineering has also been reviewed. SOCRADES requires the engineering of systems composed of loosely coupled distributed units of functionality in service-oriented architectures. This context requires capabilities to model and support the engineering of applications in a manner that can maximise the advantages of the approach. Section 14 of this document therefore provides a specific review of component-based systems engineering and details of related research projects

Ongoing tasks 7.3, 7.4 and 7.5 are seeking to define how future applications engineering tools should be implemented to meet user needs better.

2. Evaluating the enabling technologies, tools and methods

2.1. Categorisation and Format

Following on from the format adopted in D7.1 for the capture of user application engineering system capabilities, this deliverable considers and documents the capabilities of currently available tools and methods from the four perspectives highlighted in Figure 1. Typical capability evaluation criteria for each paradigm include:

- Control
 - What is the system model and method/approach adopted and is it suitable for the engineering of reconfigurable distributed modular automation systems?
 - What is the scope of the engineering capabilities provided?
 - Enterprise Integration
 - What capabilities are provided for the seamless integration of/with higher level systems e.g., ERP, MES?
 - What provision is there for the extension of business processes down to the control level?
 - Supply-Chain
 - What partners within the supply chain are supported?
 - What level of support is provided for their engineering activities and interactions?
 - Lifecycle
 - What phases of the lifecycle are supported?
 - What level of support is provided?
 - Virtual Engineering
 - Does the tool provide an editing and visualisation environment for concurrent design of machine control and mechanical layout?
-

- o What level of functionality does the Virtual Engineering function provide for a) virtual model editing and b) virtual model exploitation?

In assessing the capabilities of enabling methods and tools, the level of support provided and the degree of standardisation and openness are important criteria for comparison, based on the findings from deliverable 7.1.

2.2. Scope

The primary study here will focus on engineering tools/tech/methods suitable for sequence and interlock control in the discrete manufacturing domain in support of the associated project user-partners needs, i.e., at Jaguar.

3. Selected enabling technologies

These following sections take a more in-depth look at candidate engineering methods and tools considered by the project partners to be potentially applicable to SOCRADES. These sections try to capture i) the capabilities of these tools and ii) their strengths and weaknesses, particularly from a user perspective, together with an approximate evaluation of their capabilities based on the criteria described in section 2.

Table 1: List of investigated enabling technologies and tools

	Enabling Technology	Scope
1	V-Sim Control Validation Suits	Virtual Engineering Environment for Control Systems and Commissioning
2	Delmia Automation	Virtual Engineering Environment
3	ControlBuild	Control System Lifecycle Engineering
4	Unity Application Generator (UAG)	Control System, Configuration and Management
5	UGS Tecnomatix	Virtual Engineering Environment
6	Unity Pro	Schneider PLC Programming Environment with XML Connectivity
7	COMPAG	Process Definition Environment (PDE)

4. V-Sim Controls Validation Suits

4.1. General Description

V-Sim software¹ is developed by V-Sim Incorporated. The set of tools and functions provided by the V-Sim software focuses on the modelling of production systems and their component parts (e.g., clamping systems, multi-axis robots, conveyor sections, etc). Virtual system modelling mainly consists of 3D geometry and kinematics modelling (inverse robot kinematics), part conveying and part sensor behaviour modelling. The V-Sim workflow process relies on the implementation and use of re-usable and re-configurable model components that can be composed into complete production lines/cells [22].

¹ V-Sim software is derived from Visual Component and is very similar from a functional and architectural point of view.

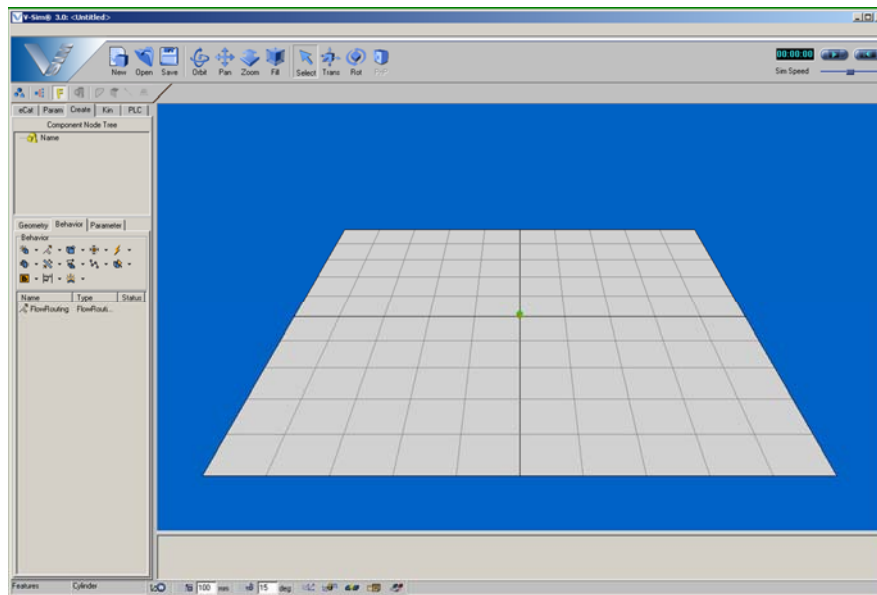


Figure 2: V-Sim User Interface

4.2. Overview of Capabilities

V-Sim supports a set of functions specifically related to the modelling of virtual components and virtual production systems/cells. Two main modelling areas can be defined: 1) 3D modelling functions that include parametric 3D geometry modelling of basic 3D objects (boxes, cubes, extrusions, etc.), kinematic joints modelling and implementation of parent-child structure between joints. 2) Behaviour modelling functions can be classified into a) complex kinematics modelling (e.g., inverse kinematics for robot motion), b) conveyor behaviour modelling (i.e. part source/sink, part queuing, part routing), c) sensor behaviour modelling (beam, volume sensor modelling, sensing behaviour), and d) generic scripting using V-Script (i.e., use of Python-based script for custom behaviour modelling).

The V-Sim workflow is based on the concept of component re-usability, i.e., systems are built by assembling components rather than by editing models from scratch. It should be noted that V-Sim provides consultancy services for building the customer's component library (which is the most time/skill consuming task).

V-Sim does not provide any tools/functions directly related to the editing/debugging of control logic or related to the production process design, management or analysis. However, interfacing with control development/testing environments is possible through an OPC server connection (includes model/PLC I/Os mapping interface).

Event log capabilities are provided, which allow basic analysis of simulation data. Information such as number of parts processed and cycle time approximation for a given machine configuration can be extrapolated. However tools dedicated to detailed process analysis, production data, and production and process management are not provided.

4.3. Summary of Strengths and Weaknesses

V-Sim has succeeded in developing a very usable application with functions precisely targeted toward the editing of virtual production cell/machine 3D models. 3D geometry, kinematics and behaviour modelling functions are accessible through an intuitive interface, which makes the software relatively simple to use while maintaining a level of functionality that enables practical system behaviour to be effectively modelled. However, the modelling of the behaviour of complex components/models (e.g., complex part management) relies heavily on V-Sim's scripting capabilities (rather than on the built-in software functions), which requires a steeper learning curve and specific programming skills.

V-Sim could be kept “simple” partially because it does not support built-in logic editing/debugging functionalities, and only provides interfacing functions to OPC servers in order to validate control code. Advantages are that V-Sim model can be tested using any logic running on OPC servers. However, practical use of the software has shown that the linking OPC server and V-Sim model required an additional process step, which was often a source of errors, and hence required further debugging.

In addition, CAD geometry import functions proved to be problematic and the use of third party CAD format translation software was required to complete the evaluation process, which has resulted in a rather complex and time-consuming workflow.

A large part of V-Sim modelling functions is dedicated to the modelling of manufactured parts, part conveying and part sensing behaviour, and implementing such behaviour represents a large amount of the total modelling time (about 40% of the total modelling time during the evaluation process). The added value of detailed part modelling can be discussed, especially regarding the time/effort required to implement this part of the final model.

4.4. Evaluation Summary

Table 2: Evaluation of V-Sim based on 4-dimensional criteria

Control		Enterprise Integration		Supply Chain		Lifecycle	
Control Capabilities	1	Business Level	0	End-user	2	Concept Design	0
Process Description	2	Application Level Interaction	2	Machine Builder	4	Design/Build/Commissioning	3
Module Definition	4	Device Level Interaction	2	Control Vendor	2	Operation/Reconfiguration	4
Virtual Engineering			4	<i>The capability is scored out of 5 (5 being the highest).</i>			

5. Delmia Automation

5.1. General Description

Delmia Automation extends the suite of Dassault System’s end-to-end Process Lifecycle Management (PLM) solution and provides an environment (functions and user interfaces) dedicated to a) the implementation of production systems 3D models, and b) the editing, testing and debugging of system control logic against the 3D model. Delmia automation differs from V-Sim by its size, complexity since it is based on Dassault’s new V5 common development platform on which all Dassault products are developed.

5.2. Overview of Capabilities

The functions supported by the Delmia module can be classified into two main groups: a) 3D/kinematics modelling, management and assembly of so called “smart devices” (equivalent to V-Sim’s model components), b) editing, testing/debugging of control logic. The former group of functions consist of a subset of Dassault’s CATIA CAD package, i.e., basic 3D geometry modelling, and assembly mating functions for defining joints’ degree of freedom. Logic editing tools enable machine internal behaviour to be described as ladder logic, function blocks or SFC+ (based on Grafset/IEC61131-3 representation) diagrams [21].

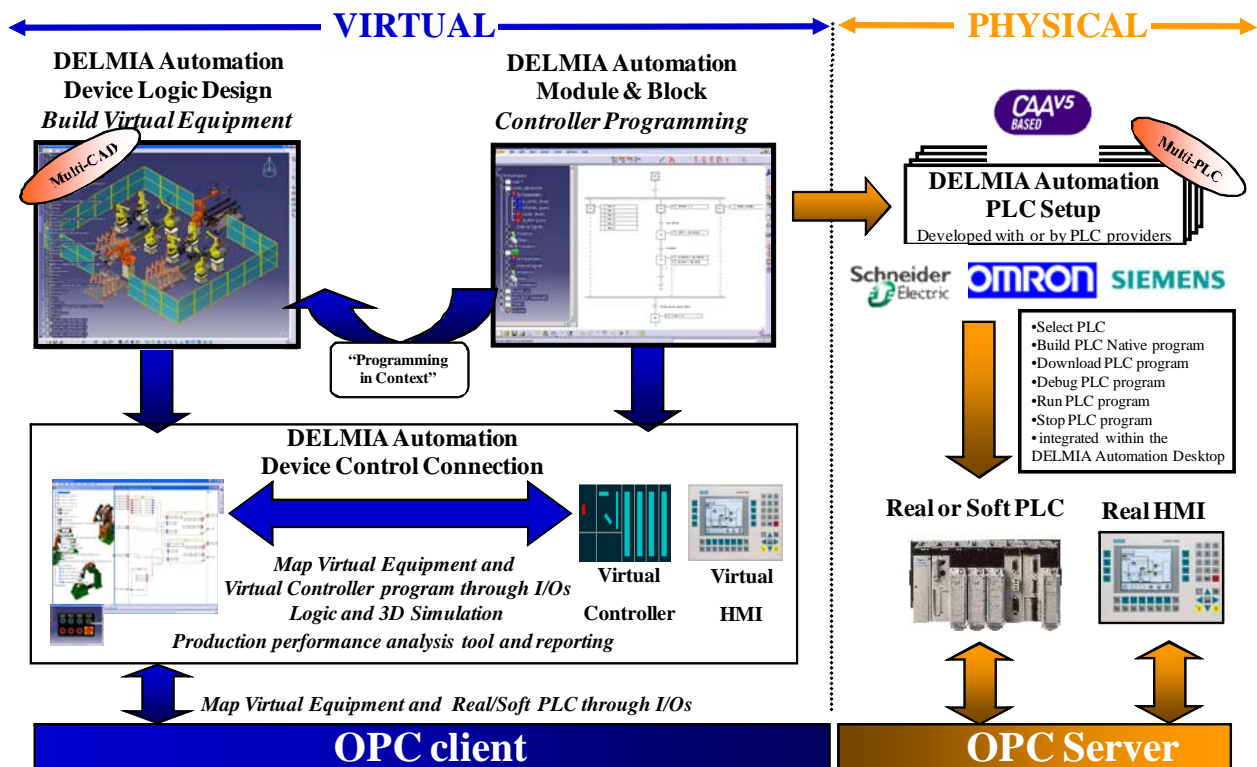


Figure 3: Delmia Automation Overview (source Delmia)

Internal machine logic can be tested and debugged against the 3D models. Control code can be generated for main PLC vendors (e.g., Siemens). Delmia Automation also provides interfacing capabilities with OPC servers, including a model to PLC (OPC) I/Os mapping interface.

Part modelling, part sensing and part routing is possible; however, this aspect of Delmia Automation software could not be assessed precisely.

5.3. Summary of Strengths and Weaknesses

Delmia Automation has been designed as a software module that will provide maximum functionality when integrated in the Dassault PLM tools suite. As an example, CATIA assembly models (including kinematics joint definition) can be directly imported into the Delmia Automation environment. In the same way, output from Delmia Automation can be used in the ENOVIA LCA module to access higher level production and process management analysis and simulation tools.

The Delmia Automation editing environment and user interface are quite complex and unintuitive. The concept of re-usable “smart devices” (i.e., modelling constructs) does not translate in a clear workflow, and relatively simple devices (e.g., clamping mechanisms) can have a highly complex internal structure and data model, which makes it difficult in practice to realise the promised benefits in terms of device reconfigurability and reusability.

Several training courses and a large amount of Delmia support was required to complete the evaluation process. It is foreseen that a large amount of training would be required in order to take full advantage of the Delmia Automation tools, which reduces the effectiveness of these tools in a mixed/non-specialised skill environment and in the early design phases.

Simulation of models created with Delmia Automation is resource heavy and typically require dedicated PCs with high-end hardware configurations.

5.4. Evaluation Summary

Table 3: Evaluation of Delmia Automation based on 4-dimensional criteria

Control		Enterprise Integration		Supply Chain		Life cycle	
Control Capabilities	4	Business Level	1	End-user	2	Concept Design	1
Process Description	3	Application Level Interaction	1	Machine Builder	4	Design/Build/Commissioning	3
Module Definition	3	Device Level Interaction	3	Control Vendor	2	Operation/Reconfiguration	4
Virtual Engineering			4	<i>The capability is scored out of 5 (5 being the highest).</i>			

6. Unigraphics eM Tool Suite

6.1. General Description

eM-Engineer and eM-PLC are part of Tecnomatix's tool set, which enables the generation of PLC logic code and provides a simulation environment for control verification. The approach is similar to Delmia and PDE tools, which is to provide the tools required to build a virtual environment from both machine control and geometry/kinematics data. This virtual environment should a) enable a better collaboration between mechanical and control engineer, b) provide simulation capabilities to test/debug machine configuration (control and mechanical) and hence detect design errors before ramp-up phases, and c) extend the use of virtual models to all engineering domains and design phases [22].

6.2. Overview of Capabilities

eM-PLC is an application jointly developed by Tecnomatix and Siemens. It is integrated with Siemens STEP 7 Professional, which is essentially an SFC programming tool. eM-PLC works in conjunction with eM-Engineer, which allows a) importing 3D CAD geometry of machine/machine parts and b) editing of kinematic data (i.e., kinematics joint definition).

Sequence of operations can be defined using drag and drop in a Gantt chart from signals for control, and a sequential function chart is automatically generated. Additional control data such as transitions conditions can be added to complete the control design. STEP 7 projects can be automatically generated, which are essentially a series of SFC diagrams. The SFCs can be tested against an in-house simulated PLC and verified using the virtual plant model. The eM environment provides interfacing to an OPC server for further testing with real hardware and/or HMI.

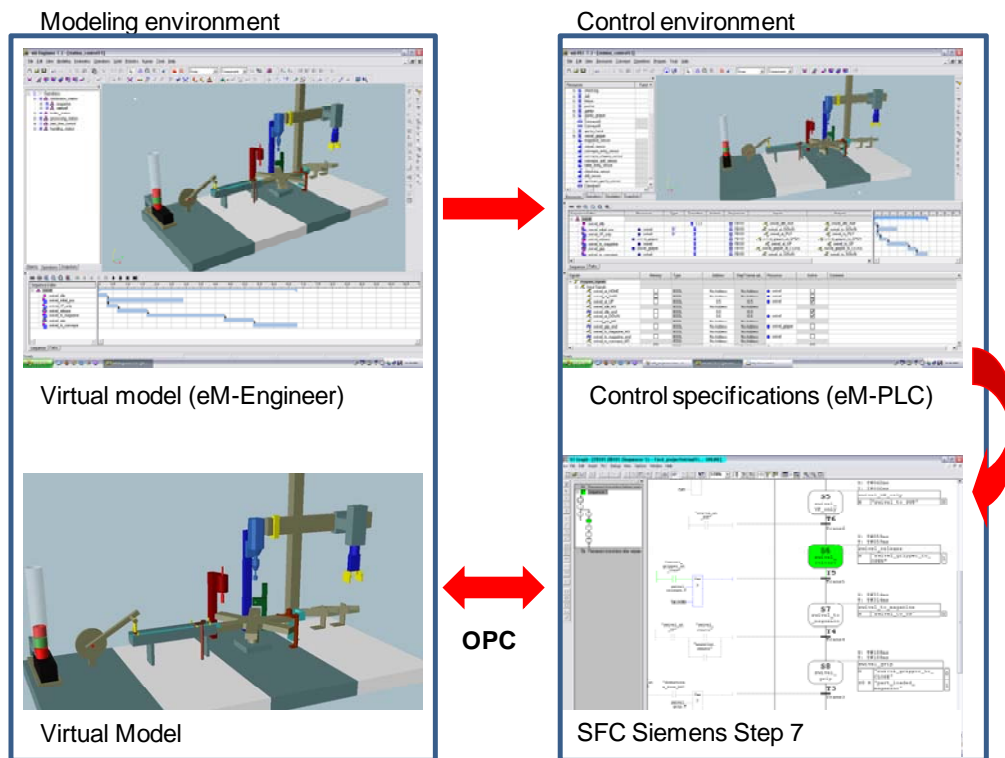


Figure 4: eM-Engineer and eM-PLC Overview

6.3. Summary of Strengths and Weaknesses

The Unigraphics eM tools suite targets the same market as Delmia Automation products. It seems that a better integration between control and 3D modelling tools has been achieved on the Dassault product thanks to the V5 platform, which, in practice, allows the kinematics modelling phase to be simplified.

It seems that the concept of re-usability through the design and use of “constructs” (e.g., components in V-Sim and PDE, Smart Devices for Delmia) is not as clearly implemented as in other applications reviewed in this document, which would practically translate into a longer modelling phase and less potential for model re-configuration.

However, eM PLC adopts an innovative approach to the generation of control logic by using Gant-like diagrams (or timing diagrams) as the starting point of the logic editing process. Sequences of operation are described and SFC plus diagrams are automatically generated. A similar feature is currently under development in the Delmia environment as a result of the demand from specific customers.

6.4. Evaluation Summary

Table 4: Evaluation of Unigraphics eM tool suite automation based on 4-dimensional criteria

Control		Enterprise Integration		Supply Chain		Lifecycle	
Control Capabilities	4	Business Level	1	End-user	2	Concept Design	2
Process Description	4	Application Level Interaction	1	Machine Builder	4	Design/Build/Commissioning	3
Module Definition	2	Device Level Interaction	3	Control Vendor	2	Operation/Reconfiguration	4

Virtual Engineering	3	<i>The capability is scored out of 5 (5 being the highest).</i>
---------------------	---	---

7. Process Definition Environment - PDE

7.1. General Description

The PDE (Process Definition Environment) tool set provides a lightweight, non-proprietary package to support production system lifecycle management enabling a) the design, visualisation, testing and debugging of control logic, b) a 3D modelling environment for building machine models against which the control logic can be tested. The PDE tool set uses standard VRML/X3D formats for 3D modelling and generic state-transition diagrams to support control logic editing/visualisation [2, 17].

7.1. Overview of Capabilities

Logic Editing and Virtual modelling environments are integrated around a common data structure referred to as common model architecture, which describes a hierarchical system as a composition of “component” and “element”. Components encapsulate modelling and control data into a re-usable construct, which can be re-configured and re-used across various applications.

The COMAPG PDE environment provides interfaces to the external environment; control-related events can be sent and received through the standard OPC client/server interface. A so-called Broadcaster enables the linking to Web-based applications (e.g., Web-based HMI, Web-based VRML models).

The modelling environment does not provide modelling tools but provides CAD geometry import capability. Kinematic modelling tools are provided, which allow defining model joints. Data such as robot TP or CNC can be imported and parsed in order to animate the machine/cell model.

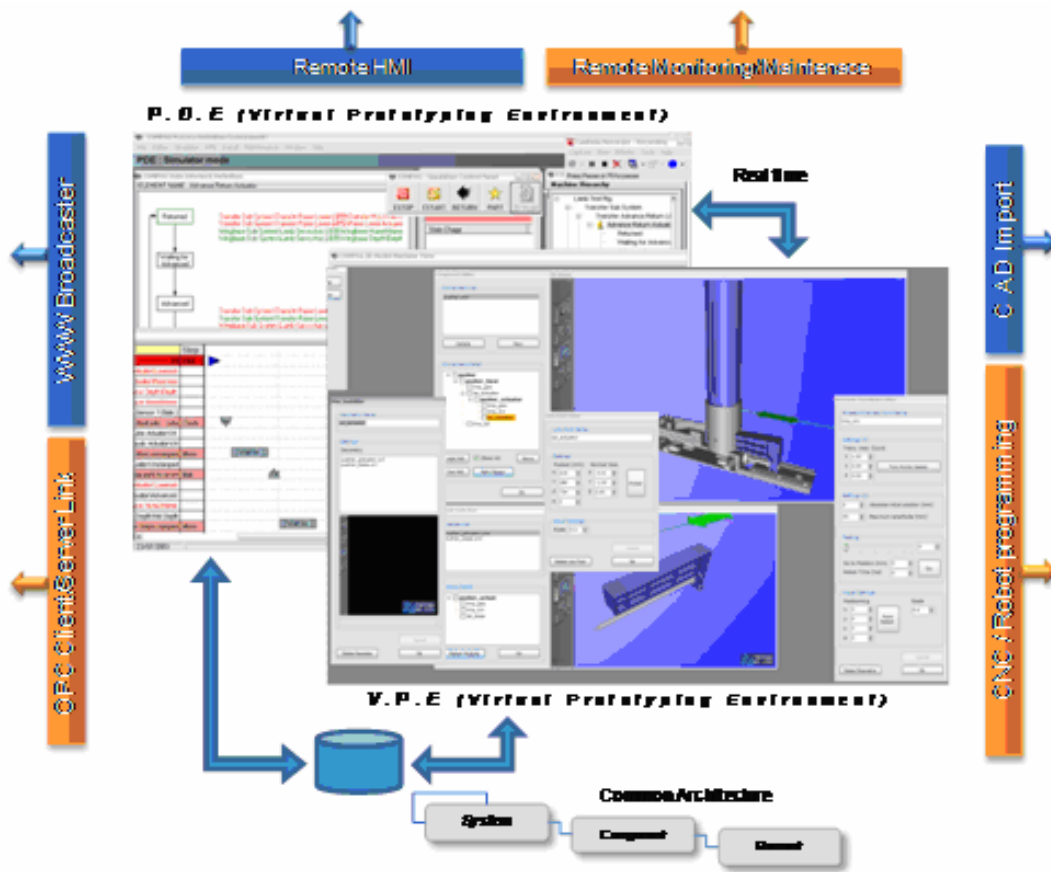


Figure 4: COMPAG PDE Overview

7.2. Summary of Strengths and Weaknesses

The concept of a component and the well-defined hierarchical breakdown of systems translate into a very clear and intuitive workflow and user interface for both the logic editing and model editing environments. The clear sequence of operation and intuitive GUI arrangement provides a high level of usability, which can potentially extend the scope of the software to the early design phases (conceptual design) and among partners with various/non-specialist skills.

The PDE logic editing environment provides a large panel of tools, views and modes that allow effective editing and testing of control logic (e.g., state transition diagram, timing/sequence diagrams, step by step mode run time). Control logic can be exported in XML format that can be used by third party converter to generates vendor specific control code. However, the PDE does not provide the level of flexibility offered by Delmia that provides a wide panel of functions for control editing (i.e. Function Blocks, Ladder logic diagrams), which for instance allows numeric control systems to be designed and tested.

The VRML (Virtual Reality Modelling Language) standard is used within the modelling environment as 3D format. This provides direct import capability from virtually any CAD package currently available, which represents an advantage compared to modelling environment based on proprietary formats. The modelling tool development is focused on simplifying the modelling task and use of features such as link points enable Lego-like assembly of component. However, the PDE modelling functions is limited to the geometry and kinematics modelling and does not include part, part routing and part sensing.

7.3. Evaluation Summary

Table 5: Evaluation of PDE based on 4-dimensional criteria

Control		Enterprise Integration		Supply Chain		Lifecycle	
Control Capabilities	3	Business Level	0	End-user	3	Concept Design	3
Process Description	2	Application Level Interaction	0	Machine Builder	3	Design/Build/Commissioning	2
Module Definition	3	Device Level Interaction	3	Control Vendor	0	Operation/Reconfiguration	3
Virtual Engineering			3	<i>The capability is scored out of 5 (5 being the highest).</i>			

8. ControlBuild Application Generator

8.1. General Description

TNI-Software, a provider of design tools and consulting services for the development, implementation and test of software dominant embedded systems, has developed ControlBuild, the first front-to-back environment for the design, development, implementation and maintenance of control systems applications. ControlBuild incorporates a host of features and functions, including C and multi-PLC automatic code generation, automatic document generation, the ability to create re-usable object libraries, a virtual commissioning capability based on the simulation and validation of systems prior to deployment and an application service and maintenance support capability [23].

8.2. Overview of Capabilities

ControlBuild is dedicated to the specific needs of control systems engineers through all phases of the application development cycle (i.e., from the definition and validation of specifications to implementation and deployment). It facilitates a holistic approach to the design and deployment of control systems applications based on an object-oriented design approach.

ControlBuild includes an automatic code generation, network, communication parameters and more capability for C, multiple programmable logic controller (PLC) platforms including Schneider's PL7-Pro and UNITY-Pro, Siemens' STEP7 and XML using the PLCopen format. ControlBuild also provides direct visualisation of code during the design phase, a native code generator allows control engineers to switch easily between the different IEC 61131-3 standardised PLC programming languages (i.e., Structured Text (ST), Function Block Diagram (FBD), Ladder Diagram (LD), Instruction List (IL) and Sequential Function Chart (SFC)).

ControlBuild enables also the automatic exportation to HMI/SCADA tag data to most of the solution; based on generic XML format export files the user can easily adapted it to other tools.

8.3. Summary of Strengths and Weaknesses

ControlBuild is a tool set oriented towards component-based control system software generation, configuration and support rather than virtual engineering.

8.4. Evaluation Summary

Table 6: Evaluation of ControlBuild based on 4-dimensional criteria

Control	Enterprise Integration	Supply Chain	Lifecycle
---------	------------------------	--------------	-----------

Control Capabilities	4	Business Level	1	End-user	1	Concept Design	0
Process Description	1	Application Level Interaction	2	Machine Builder	1	Design/Build/Commissioning	2
Module Definition	3	Device Level Interaction	3	Control Vendor	4	Operation/Reconfiguration	3

The capability is scored out of 5 (5 being the highest).

9. Unity Application Generator (UAG)

9.1. General Description

Unity Application Generator (UAG) is a process design tool for process automation developed by Schneider-Electric. The goal is to offer the user a single tool to design a distributed automated solution and targets at closing the gap that exists between process engineers and control engineers. The designer uses components to create the automation system in an ISA S88 standardized model and maps the control behaviours of the components to resources/control hardware, which build the control architecture of the system [24].

9.2. Overview of Capabilities

UAG provides preinstalled libraries with Smart Control Devices (SCoDs) or Control Module Types, which are instantiated then as Control Module in a physical process model. A SCoD can be perceived as a real automation object like a valve, motor or tank. It integrates all necessary artefacts like the PLC logic, animated HMI symbols, documentation, access-logic for manual HMI intervention, HMI alarming and more.

For the process engineer UAG allows the user to define a general layout of the process based on objects defined in the Physical Model of ISA S88.01 standard like Area, Process Cell, Unit, Equipment Module and Control Module and to link from UAG objects to his basic tools like E-plan, Autocad, P&ID drawings etc.

For the control engineer UAG allows the user to build the control architecture with PLCs, HMIs and networks as the Topological Model, to assign the control logic to the objects the process engineer has defined and to generate a major part of the control logic for the PLC and the HMI from the process design. After generation control code can be downloaded to the PLC or the HMI.

9.3. Summary of Strengths and Weaknesses

UAG follows a component-based approach, which speeds up the design phase and promotes reuse of existing and tested parts of the system. Simulation, advanced verification and debugging of the generated logic have to be provided by the external targeted tools. Kinematics and geometry of the components are not respected by UAG; these can be provided at most as additional linked documentary material.

Based on the physical and the topological structure, UAG is able to generate the skeleton of the automated solution including configuration of hardware, network, communication parameters and more. The generated projects have to be refined and completed in the necessary external engineering tools, as the behavioural part is incomplete. Control Module interactions can be realized through interlock or direct links, but are mostly not sufficient to model the overall behaviour.

User-defined generators can be developed by third parties to extend the capabilities of UAG. It provides a documented API to write plug-ins that are able to manage additional applications. Currently, UAG generates projects for Schneider-Electric PLCs Quantum and Premium, and some HMI/SCADA solutions like Vijeo Citect, Monitor Pro, iFix or generic XML export files that can be transformed and digested by other supporting tools.

UAG principally supports the “incremental generation”, which means it does not simply overwrite the previously generated project, but applies only the modifications. However, if incremental builds work depends on different factors, such as the complexity or openness of the project database structure, or, in case an API has to be used if such incremental operations are supported. Therefore UAG is not only applicable at the design phase but also at later stages, for instance at reconfiguration time.

9.4. Evaluation Summary

Table 7: Evaluation of UAG based on 4-dimensional criteria

Control		Enterprise Integration		Supply Chain		Lifecycle	
Control Capabilities	3	Business Level	1	End-user	0	Concept Design	0
Process Description	3	Application Level Interaction	2	Machine Builder	1	Design/Build/Commissioning	3
Module Definition	3	Device Level Interaction	4	Control Vendor	3	Operation/Reconfiguration	3

The capability is scored out of 5 (5 being the highest).

10. Unity Pro

10.1. General Description

Unity Pro is the common IEC 61131-3 programming, debugging and operating software developed by Schneider-Electric. Goal is to simplify and unify the programming environment for different PLC families of Schneider-Electric like the Quantum, the Premium and the upcoming M340. UnityPro is the unique successor of the two formerly used PLC programming tools (Concept and PL7).

Although the range of the supported PLC is limited, Unity Pro provides openness and interoperability to other engineering tools via published s/w automation interfaces and an open XML-based exchange format [25].

10.2. Overview of Capabilities

Five IEC 61131 languages compliant to IEC 61131-3 (*LD, FBD, SFC, IL, ST*) are supported with PLC specific optional language extension. This way PLC code can be exported from Unity Pro to be PLCopen-compliant.

The application can be structured in two ways: a) an application can be decomposed into tasks, sections, operator screens and animation tables; b) an application can be structured functionally into hierarchical functional modules.

The operational phase of a production system is supported by allowing Unity Pro to monitor variables at runtime. Online access to process data and visualization can then be done in animation tables and operator screens mentioned before.

Apart from that Unity Pro allows graphical animation of IEC editor sections for debugging purposes, so that logical states or values are presented appropriate for the different IEC languages. Debugging is supported by allowing the tool to set break- or watch-points in the PLC.

A PLC simulator allows running and testing parts of the application without physical presence of a PLC.

Unity Pro follows a client/server architecture in which a server (P-Server) provides interfaces for external access to the Unity data model and operations. Unity servers are based on COM/DCOM and OLE automation so that external clients can be programmed in C#, C++, VB and VBA.

10.3. Summary of Strengths and Weaknesses

The capabilities are aligned to the demands of PLC programming rather than to demands of distributed system programming. Unity Pro is not a PLC platform-agnostic programming tool, it is clearly focussed on specific PLC families. This allows utilization of specific performance-relevant features such as hot-standby, multi-tasking capability or different task priorities (event-triggered, high- and low-priority for instance), which is probably hard to reach in generic programming tools.

The strength of Unity Pro is the openness to be accessed by 3rd party tool vendors, so that Unity can be part in a greater engineering context.

One tool instance manages one PLC, which means that decentralized control systems cannot be operated and debugged by one tool instance.

10.4. Evaluation Summary

Table 8: Evaluation of UnityPro based on 4-dimensional criteria

Control		Enterprise Integration		Supply Chain		Life cycle	
Control Capabilities	5	Business Level	0	End-user	1	Concept Design	1
Process Description	4	Application Level Interaction	1	Machine Builder	1	Design/Build/Commissioning	4
Module Definition	3	Device Level Interaction	4	Control Vendor	4	Operation/Reconfiguration	3

The capability is scored out of 5 (5 being the highest)

11. SAP

11.1. General Description

The SAP xMII Manufacturing Integration and Intelligence, is defined as a Manufacturing Intelligence Portal™ that promotes global and distributed e-manufacture by: a) Web-enabling all existing manufacturing data and information systems, b) XML-enabling all plant floor systems and information to simplify integration into e-business applications such as ERP, CRM and c) Enabling real-time views of the supply chain data and status to support decision making processes [26].

The SAP xMII's functions can be broken down as follow: a) extraction of data from multiple data sources, b) aggregation and processing of data into a business context, c) dissemination of results to the user into specific/user defined forms. Support for specific information infrastructure and technologies are provided (i.e. Wireless medium, Web-based services, Client/server services).

11.2. Overview of Capabilities

The SAP xMII provides an environment to allow users to correlate data from multiple systems via a simple browser interface. The software does not need to be managed at the client site and no requirement for complex data warehouses or data models exist with an SAP xMII implementation.

Architecture: The SAP xMII was designed as an N-tier application that is platform and browser neutral. Its basic architecture can be divided into three sections: the data source layer; the application layer, and the user layer consisting of display components such reports or dynamically-generated XML content. All of the SAP xMII functionality is exposed using a standard Web services-based architecture via uniform resource identifier (URI) requests. .

Application Layer: The application layer is based on a standard Web server with a servlet engine (Java servlet technology). The server side functions are designed to get data, transform it, and deliver it to any requesting client.

Connectors: Connectors integrate any plant focused data source into the SAP xMII. The connectors provide data collection, conversion, aggregation and abstraction. All data is abstracted as XML data streams.

Product Operation: Connectors run on the Web server and respond to authenticated requests for information from a client. The connector protocol layer manages the request and returns the appropriate data in the requested output format. The SAP xMII uses standard HTTP/HTTPS, allowing it to work seamlessly with corporate firewalls and leverage Web and Web server security.

Protocol Layer: The protocol layer manages all of the communication to the plant data sources.

Data Rendering Layer: The rendering layer abstracts the data source from the data presentation. Current outputs and their typical uses include HTML, SAP xMII Applets, XML, Server-Side Images etc

Productivity Tools: A Template Editor allows constructing a query against a data source and presenting the data into specific/user-defined formats. A Dynamic Page Generator allows end users to create a dynamic HTML pages. An add-in for Microsoft FrontPage and Macromedia Dreamweaver provides both an easy way to construct applets and images from SAP xMII data sources.

SPC Data Analysis: SPC data analysis provides visual feedback of data analysis and related events, to users. Data from existing data sources are into an SPC calculation engine. The resulting calculations are delivered in a view through the SPC applet library, allowing for alert notifications of rule violations through email and other messaging services. A framework allows calculation results to be stored/logged.

11.3. Summary of Strengths and Weaknesses

The main strength of SAP xMII is in the integration of the manufacturing shop-floor with enterprise services. xMII is able to extract data from multiple data sources, aggregate the data at the server, transform it into a business context, and personalize and deliver results to users.

xMII was not designed to provide sophisticated control capabilities and should not be used as such. SAP environment support for shop floor is currently available only using UDS.

11.4. Evaluation Summary

Table 9: Evaluation of SAP based on 4-dimensional criteria

Control		Enterprise Integration		Supply Chain		Lifecycle	
Control Capabilities	0	Business Level	5	End-user	4	Concept Design	5
Process Description	4	Application Level Interaction	5	Machine Builder	4	Design/Build/Commissioning	5
Module Definition	0	Device Level Interaction	4	Control Vendor	4	Operation/Reconfiguration	5

The capability is scored out of 5 (5 being the highest).

12. Catalogue of Investigated Technologies

Table 10: Summary of evaluated enabling methods and tools

Enabling Technologies and Tools		V-Sim	Delmia	UGS	Compag	Control Build	UAG	Unity Pro	SAP
Control	Control Capabilities	0	4	3	3	4	3	5	0
	Process Description	0	1	3	2	1	3	4	4
	Module Definition	4	3	2	3	3	3	3	0
Ent. Integ.	Business Level	0	0	1	0	4	4	5	5
	App. Level Interaction	1	1	1	0	0	0	0	5
	Device Level Interaction	3	3	3	3	3	3	3	4
Supply Chain	End-user	1	2	2	3	1	1	0	4
	Machine Builder	3	4	4	3	2	2	1	4
	Control Vendor	0	2	2	2	3	4	4	4
Life Cycle	Concept Design	1	2	2	3	1	0	1	5
	Design/Build/ Comm.	2	2	2	2	2	2	2	5
	Operation/Reconfig.	4	3	4	3	4	3	4	5

The capability is scored out of 5 (5 being the highest).

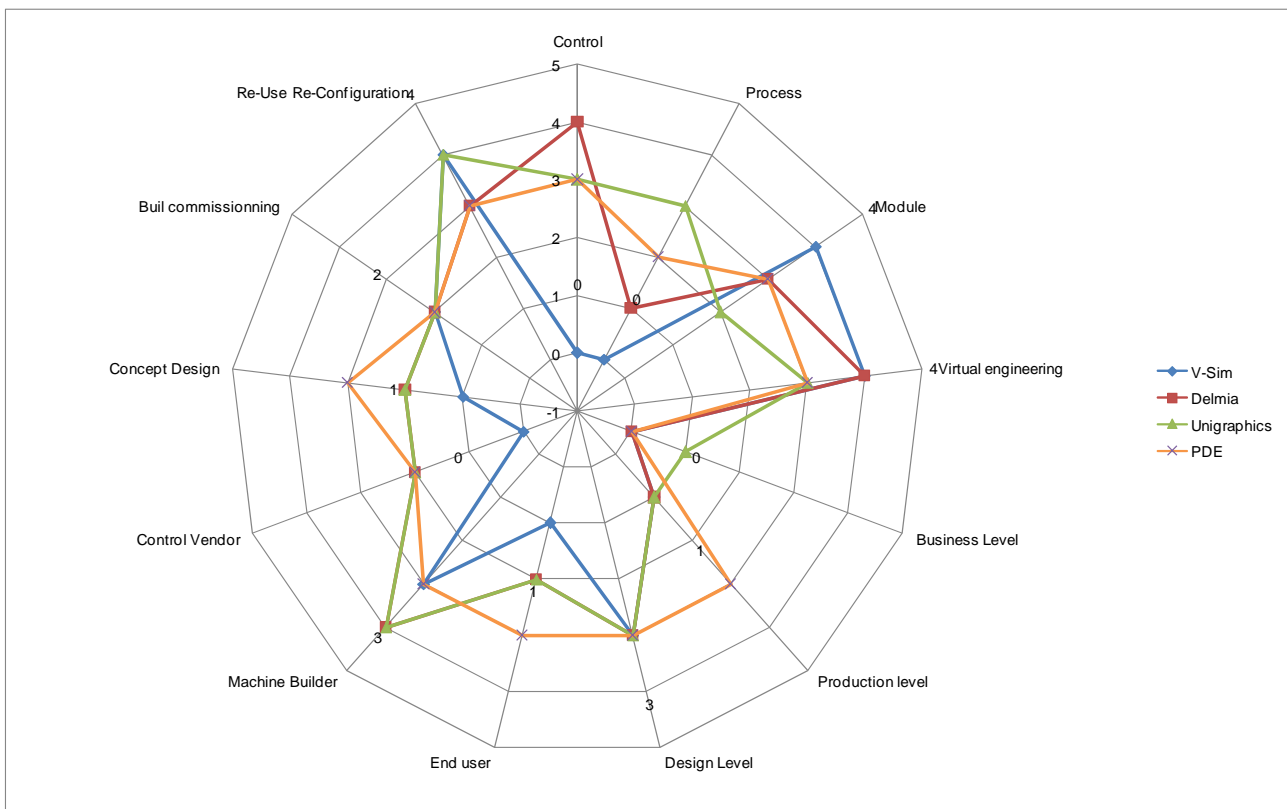


Figure 5: Coverage of Virtual engineering enabled technologies (Best seen in colour)

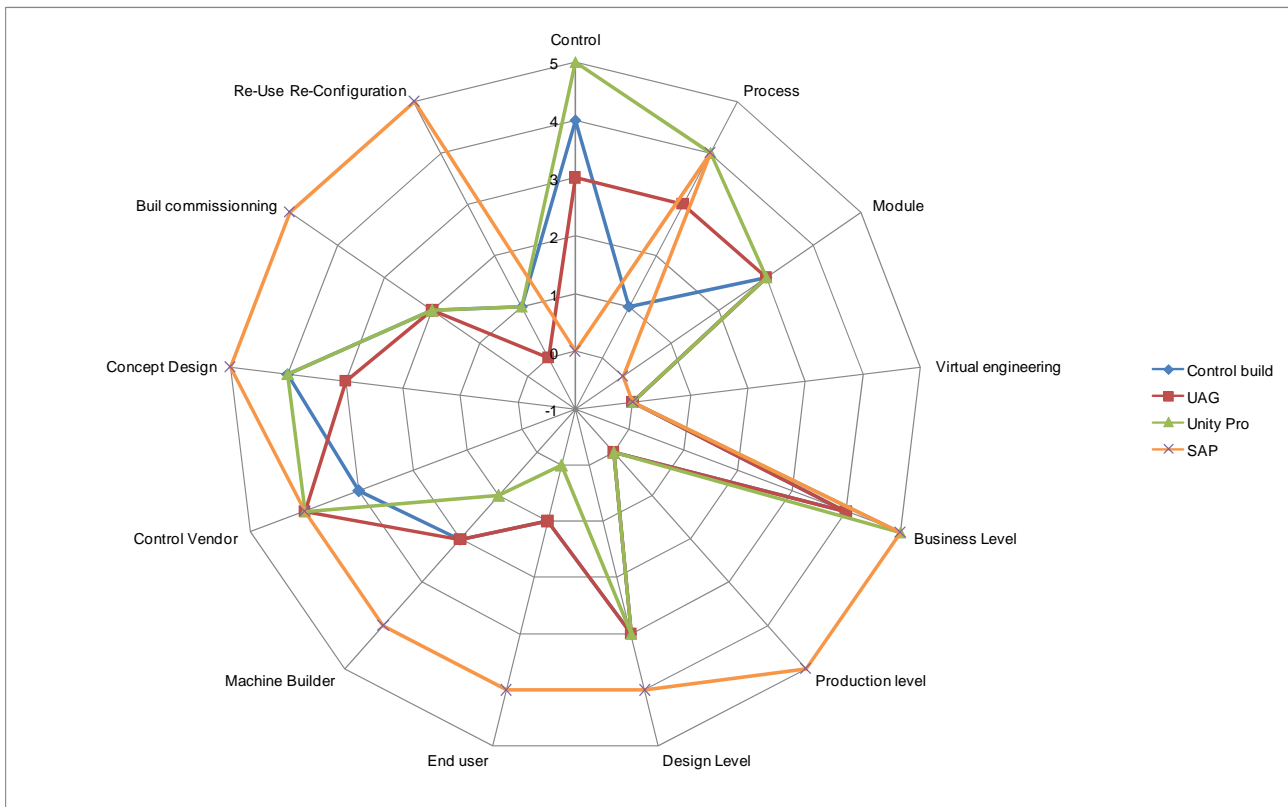


Figure 6: Coverage of Control specific technologies (Best seen in colour)

13. Evaluation of Reviewed Engineering Tools

The set of tools reviewed in this document can be classified into two main categories i) software tools sometimes referred to as “virtual engineering,” which introduce a set of functionalities related to the design and implementation of a 3D-based virtual model, and ii) environments with functions oriented towards control software editing and control code generation.

Software tools for “virtual engineering,” (VE) aim to provide intuitive views of the system being designed where the main objective is to break down the barriers that typically exist among control, process and mechanical engineers. This approach raises several software design issues including the editing and manipulation of 3D data and the integration between 3D modelling and control data and events. In the implementation of functions and user interfaces related to the editing and manipulation of 3D data the main challenge is to implement tools that are accessible and effectively usable to non-3D-specialists. As an example, Delmia seek to achieve this goal by enabling direct import from its proprietary CAD package. On the other hand, tools such as COMPAG PDE achieve this goal by enabling direct import through standard 3D formats. Achieving effective integration between 3D modelling and control data is important so that a model’s behaviour can be consistently maintained as a combination of control logic and dynamic model motions in order to conduct process and sequence analysis. Integration of 3D and control related data require an additional step in the system engineering process and therefore need to be made as transparent to the user (i.e., engineers) as possible. This can be achieved through the use of an integrated database system (e.g., Delmia) or a common data model (e.g., PDE COMPAG). Tools such as V-Sim do not provide control design functions but achieve simulation capabilities through the use of an OPC interface, although the integration between model and external logic in a non-integrated environment cannot be tightly controlled/automated and is a potential source of errors.

Environments with functions oriented towards control software editing and control code generation aim at assisting control and process engineers to manage the task of implementing control code for complex/large-scale systems by a) providing direct translation of process production sequence data into usable code,

potentially in distributed form, at the embedded control hardware level and b) providing means to maximise re-use and re-configuration of this control code.

Each of the above types of environment is traditionally targeted at different end-users (e.g., machine builders/machine customers for a VE-based environment, control vendors/machine builders for a control-specific environment) and different phases of the system lifecycle (i.e., concept design for a VE-based environment, build/commissioning for a control-specific environment). Both types of environments can provide potential benefits in the system re-configuration phase.

SOCRADES requires the engineering support for systems composed of loosely coupled, distributed units of functionality in a service-oriented architecture. This context requires capabilities to model and support the engineering of applications in a component-based manner, which can maximise the advantages of the approach. This topic is addressed in the next section.

14. Service-Based and Component-Based Systems Engineering

The systems engineering of distributed embedded systems requires the modelling and support of the units of distributed functionality. Object-based specifications emphasize structural decomposition, which facilitates the implementation of open and reconfigurable systems, e.g. industrial software standards such as IEC 61131-3 [12] and IEC 61499 [13]. In that case the system is conceived as a composition of interacting components, such as function blocks and port-based objects, which are then mapped onto real-time tasks. Object-based design uses a number of fundamental principles such as encapsulation, aggregation and association of objects (components). A major problem that has to be overcome is the informal and largely ad-hoc definition of application objects.

This can be observed in many software design methods, where it is left to the system designer to define system objects and object classes for each particular application. That applies not only to component functionality and interfacing, but also to the way components are mapped onto real-time tasks. It can be argued that a software design method should incorporate a hierarchical model, taking into account both the structural and computational aspects of system architecture [32-36]. Such factors need careful consideration in the realisation of the SOCRADES system architecture and middleware.²

Ad-hoc specification and design severely limit component reusability. Therefore, it is highly desirable to develop a formal framework that will allow for a systematic specification of reconfigurable components, which will be reusable by definition. The specification of suitable SOCRADES components and a corresponding engineering environment to support these components is required.

Distributed applications require components -- function units such as sensors, controllers, actuators, operator stations, etc. These function units must interact transparently with one another via appropriate services to choreograph or orchestrate activities appropriately.

Component interaction is another major issue that has to be resolved while specifying system configuration. There are a number of interaction patterns, which are widely used in the context of both process-based and object-based systems: client-server, producer-consumer(s) and publisher-subscriber. Client-server is highly popular in the IT community and is also used in industrial computer systems and protocols. However, it has limitations, such as the blocking nature of communication and point to-point interactions. Therefore, producer-consumer is often considered better suited for real-time systems because it is non-blocking and supports one-to-many interactions [7]. Publisher-subscriber combines the useful features of the former two methods, and it is also widely used in practice. Web-services offer a particular set of component interaction

² There are a number of emerging approaches being put forward for the implementation of component-based distributed real-time systems. For example, research in component-based real-time systems includes the international research efforts: MoBIES, AIRES, ViPER, PECOS, OROCOS, OSACA, GenoM. The implementation of the above real-time systems is not based on Web services; however, the component-based approaches they adopt are relevant to loosely coupled systems that might typically be implemented with Web services on the SOCRADES project, and these may be relevant to the work on the underlying system architecture being implemented.

capabilities, which need to be appropriately configured by the engineering system to best effect in the context of application requirements.

In this context a number of research projects are undertaking relevant work on distributed automation systems, which is briefly reviewed in the following sections.

14.1. TORERO

The TORERO aims at creating a total lifecycle Web-integrated control design architecture and methodology for distributed control systems in factory automation. TORERO focuses on the development of integrated devices (so called TORERO Devices TDs), embedding drives and sensors as well as control code, auto-diagnostic and configuration capabilities. TDs can be connected via industrial networks and therefore can compose a distributed control network of mechatronic devices [27].

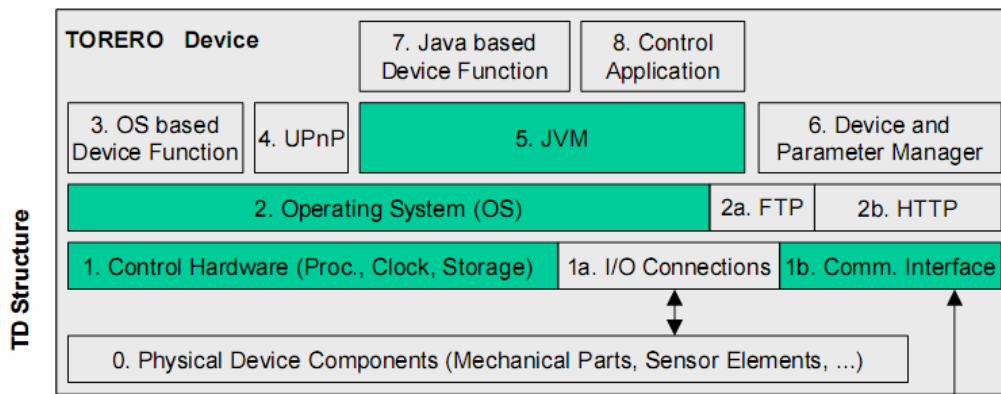


Figure 7: Description of a TORERO Device

The definition of TD is shown in the figure above. A set of APIs enable control vendors to interface a control editing environment to TD via a Java-based interface embedded in each TD, enabling the access to TDs' function and internal data. However, no details of any design environment or user interface implemented as part of the TORERO project were available, which could define a methodology and workflow to compose TDs into complete systems.

14.2. OCEAN

The main target of the OCEAN project is the definition and realisation of a "Distributed Control System Real-Time Framework" (DCRF) for numerical controls based on standardised communication systems. The DCRF should enable the hosting of control components in distributed open platforms and should provide a real-time communication API. The DCRF implementation is based on open source components (RT-Linux as operating system, RT-CORBA for standardised communication mechanisms) and will be available (open source) at the end of the project [28].

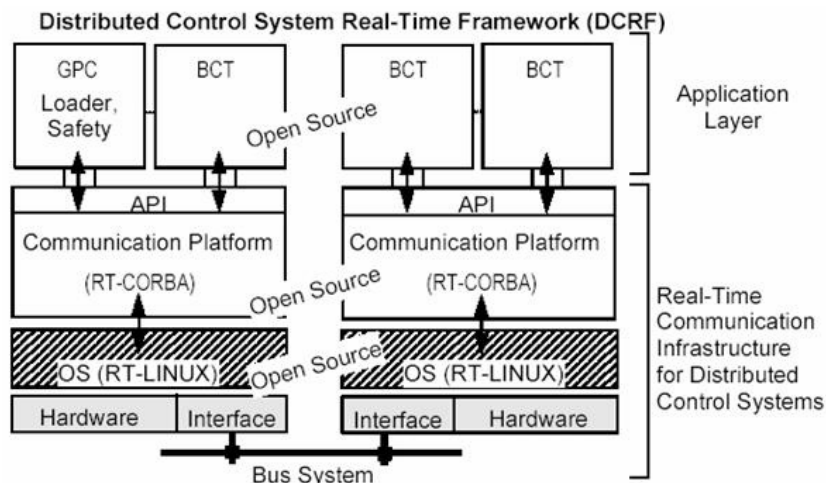


Figure 8: OCEAN Infrastructure overview

The OCEAN project therefore focuses on the implementation of a real-time operating system, which cannot be assessed with regards to the tools reviewed in this document.

14.3. SIRENA

The SIRENA project aims at promoting the development of Service Oriented Architectures (SOA) across different application domains (namely the industrial, telecommunication, automotive and home automation domains). The implementation focuses on the definition of Devices Profile for Web Services (DPWS), which will provide the interface through which devices' services are described/accessed/found/subscribed [29].

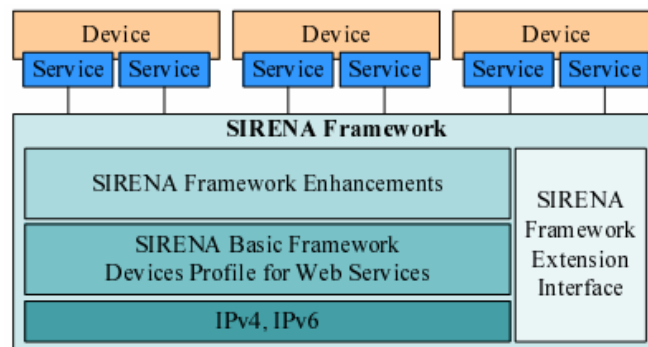


Figure 9: Overview of SIRENA Framework

The SIRENA framework provides specific services such as event management, service discovery, server request management, Bluetooth devices communication etc. and is a key enabling project for SOCRADES.

14.4. ABAS

The type of agent-based manufacturing systems that are developed using the framework employ the Actor-based Assembly Systems (ABAS) reference architecture (Martinez Lastra, 2004). ABAS are reconfigurable systems built by autonomous mechatronic devices that deploy auction- and negotiation-based multi-agent control in order to collaborate towards a common goal, the accomplishment of assembly tasks. These assembly tasks are complex functions generated as a composition of simpler activities called assembly operations, which are the individual goals of the actors. ABAS tools are used to compose an engineering framework [31].

15. Overall Conclusions

None of the reviewed engineering tools adequately support control, enterprise and supply-chain integration. Current tools are typically either focused on support of virtual engineering or focused on software development support.

The paradigm of service-oriented systems provides the opportunity to create systems naturally from loosely coupled distributed components. The creation of suitable configuration and support tools for SOCRADES will be heavily interrelated with the underlying system architecture. The engineering environment thus needs to provide a suitable approach for the specification, configuration, and behaviour of such components and a method of storing these for reuse. None of the candidate engineering tools looked at currently support Web services although ControlBuild, Delmia Automation, and PDE already offer varying degrees of support for component-based systems.

Providing a visualization environment to assist the end-user and enabling visualizations to be linked to the physical control components are important engineering requirements needed by end-users. Therefore, this study has focused heavily on examining available engineering tools with this capability.

16. References

- [1] K. Sando, G. Corbitt, R. Boykin, 2001. Enterprise integration, Published by John Wiley & Sons Inc., ISBN 0-471-35993-9.
- [2] R Harrison, A. A. West, R H Weston, R P Monfared, 2001. Distributed engineering of manufacturing machines. Proc Instn Mech Engrs, Part B. vol. 215, pp. 217-231.
- [3] Z. Xu, Z. Zhao, R. W. Baines, 2000. Constructing virtual environment for manufacturing simulation. International journal of Production research, vol. 28, no. 17, pp. 4171-4191.
- [4] J. Adolfsson, A. Ng, P. Olofsgard, P. Moore, J. Pu, C-B. Wong, 2002. Design and simulation of component-based manufacturing systems. Mechatronics, vol. 12, pp. 1239-1258.
- [5] A. Harrison, 1997. From lean to agile manufacturing, IEE colloquium, (digest), no. 386, pp. 1-7.
- [6] K. Cheng, P. Y. Pan and D. K. Harrison, 2001. Web-based design and manufacturing support systems: implementation perspectives. International Journal of computer integrated manufacturing. vol. 14, no. 1, pp. 12-27.
- [7] A. Arbor, 2002. Report on Re-configurable manufacturing systems (RMS). Engineering centre for RMS, University of Michigan. A National Foundation Engineering Research Centre.
- [8] C. G. Burdea, 1999. The synergy between virtual reality and robotics. IEEE Transaction on robotics and automation, vol. 15, no. 3, June 1999.
- [9] G. Sutcliffe, K. Deol Kaur, 2000. Evaluating the usability of virtual reality user Interfaces. Behaviour & Information Technology, 2000, vol. 19, No. 6, 415-426.
- [10] H-J. Bullinger, R. Breining, W. Bauer, 1999. Virtual Prototyping State of the art in production design. Internal report, University of Stuttgart.
- [11] R. H. Weston, 1998. Agile System Design and Built, toward next generation of manufacturing. IEE workshop: Responsiveness in manufacturing, London

-
- [12] L. M. Camarainha-Matos, H. Afsarmanesh, R. J. Rabelo, 2003. Infrastructure development for agile Virtual enterprises, *International Journal of Computer Integrated Manufacturing*, vol. 16, no. 4-5, pp.235-254.
- [13] Ravi Chwala, Amarnath Banerjee, A Virtual Environment for simulating manufacturing operation in 3D, December 2001. Proceedings of the 33rd conference on Winter Simulation WSC'01, December 2001, IEEE Computer Society
- [14] R. P. Monfared, A. A. West, R. Harrison, R. H. Weston, 2002. An implementation of the business process modelling in the automotive industry. *IMechE, Proc. Instn. Mech. Engrs.*, vol. 216, part B, pp. 1413-1427.
- [15] A. Gunasekaran, Y. Yusuf, 2002. Agile manufacturing : A taxonomy of strategic and technological imperatives, *International Journal of Production Research*, vol. 40, no. 6, pp. 1357-1385.
- [16] H. Sharifi, Z. Zhang, 2001, Agile manufacturing in practice: Application of a methodology, *International journal of operation & Production Management*, vol. 21, no. 5/6, pp.772-794.
- [17] A. A. West, R. Harrison, C. D. Wright, A. J. Carrott, 2000. The visualisation of control logic and physical machine elements within an integrated machine design and control environment. *Mechatronics*, no. 10, pp. 669-698.
- [18] R. Carey, 1997. The annotated VRML2.0 Reference Manual. Addison-Wesley. Inc. T.V.C., the Virtual Reality Modelling Language International Standard ISO/IEC 14772-1.
- [19] D. A. Vera Innovative Approach to the Design and Realisation of a Virtual Prototyping Environment for Manufacturing Systems Engineering. Phd Thesis, Loughborough University, Wolfson School of Manufacturing Engineering, UK, 2004
- [20] © V-Sim Incorporated: Virtual Manufacturing and software Emulation: White Paper, © V- Joe Hujan, , October 2006, Novi Detroit, Michigan. www.v-sim.com.
- [21] DELMIA V5 Automation Platform: Merging Digital Manufacturing with Automation, ARC White Paper February 2006, www.3ds.com
- [22] UGS Tecnomatix: The importance of Design Process, *Manufacturing Today Magazine*, March 2005, no 131.
- [23] ControlBuild, TNI Software Incorporation, www.tni-software.com/fr/produits/controlbuild.
- [24] Schneider electric Telemecanique, Automation and Control: Unity Application Generator, www.telemecanique.com/en/products/index_fon11_fam30_prod_onestep.htm
- [25] Schneider electric Telemecanique, Automation and Control: Unity™ Pro Programming, Debugging and Operating Software, IEC 61131-3 application development software,
- [26] SAP, global enterprise management and business solutions, <http://www.sap.com/solutions/index.epx>
- [27] Aspect-Oriented of Control Application Code for Distributed Automation Systems: The TORERO Approach, Springer Berlin / Heidelberg, ISSN 0302-9743, p. 335-345, October 2003.
- [28] OCEAN project: Real-time capable platform for distributed control applications, IST, Information System Technologies, European Project: http://www.fidia.it/english/research_ocean_fr.htm
- [29] SIRENA Project: Service oriented paradigms in industrial automation, ITEA Information Technologies for European Advancement, ITEA 02014. Francois Jammes / Harm Smit, Innsbruck, 17 February 2005
- [30] Bohn, H. Bobek, A. Golatowski, F. , SIRENA - Service Infrastructure for Real-time Embedded Networked Devices: A service oriented framework for different domains, April 2006 p. 43- 43, ISBN: 0-7695-2552-0
- [31] Lastra, J. L. (2004). Reference Mechatronic Architecture for Actor-based Assembly Systems. PhD-Thesis, Tampere University of Technology, Finland.
-

- [32] John, K.H., Tiegelkamp, M.: IEC 61131-3: Programming Industrial Automation Systems. Springer (2001)
- [33] Lewis, R.: Modeling Control Systems Using IEC 61499. Institution of Electrical Engineers (2001)
- [34] Wang, S., Shin, K.G.: An Architecture for Embedded Software Integration Using Reusable Components. Proc. of the International Conference on Compilers, Architecture, and Synthesis for Embedded Systems, San Jose, CA (2000)
- [35] Wagner, F., Wolstenholme, P.: Modeling and Building Reliable, Reusable Software. Proc.of the 10th IEEE International Conference and Workshop on the Engineering of Computer-Based Systems (2003)
- [36] Isovich, D., Norström, C.: Components in Real-Time Systems. Proc. of the 8th International Conference on Real-Time Computing Systems and Applications, Tokyo, Japan (2002)
- [37] A Software Engineering Approach to Design and Development of Semantic Web Service Applications, M Brambilla, I Celino, S Ceri, D Cerizza, E Della ... - Proceedings of the 5th Int. l Semantic Web Conference ISWC 2006.