SUPPORT OF CONTROL APPLICATION DESIGN USING DIGITAL DESIGN AND PLANING OF MANUFACTURING CELLS

Christian Diedrich*, Günter Franz, Karl-Heinz John****
Jan Krause*, Frank Poignee***

** ifak Magdeburg, Department of control systems,*
Steinfeldstr. 3, 39179 Barleben-Germany, Phone.: +49-(0)39203-81024, info@ifak-md.de
*** AUCOTEC Bremen GmbH*
Farenheitstr. 11, 28359 Bremen-Germany, Phone: +49 (0) 421 201163, info@aucotec-bermen.de
**** Infoteam Software GmbH,*
Am Bauhof 9, 91088 Bubenreuth-Germany, Phone: +49 (0) 9131 78000, info@infoteam. de

Abstract: The paper deals with automation aspects of digital planning processes. It describes the integration procedure of controller and SCADA systems into digital planning and simulation. This integration is an extension to the current situation where the simulation is organised within the planning tools and the generation of programs is restricted to robots. The generation of controller programs and SCADA system configuration are described in detail. *Copyright © 2005 IFAC*

Keywords: control engineering, integration, data models, planning, programmable logic controllers, validation, simulation

1. INTRODUCTION

Shorter innovation cycles and the increase of variants and flexibility of products makes the engineering process a focus of interest. One approach to reducing the engineering cost is a full digital planning and simulation of the production line. Concepts, concrete process sequences and the necessary resources can be defined and approved before the first construction process is initiated. This activity belongs to digital factory or digital planning. The main idea is to design a non-interrupted engineering process based on the already existing CAx data of the products and the equipment.

Several phases can be differentiated in production engineering: concept planning, rough planning, detailed planning, and start-up of production (see Figure 1). The task distribution usually takes place between the OEM (original equipment manufacturer), the main contractor for the complete system, and, if need be, further engineering service providers. The OEM provides the design of the product (e.g. a car door), the main contractor designs the production steps, i.e. the production process, and provides the production resources (e.g. a robot cell). For some specific resources or tasks the main contractor uses service provider (e.g. conveyor belt or safety controller). The first phase, concept planning, aims at developing a rough system concept on the basis of the results of the product development phase (e.g. geometrical product description, parts list, engineering bill of materials) and the requirements of the sales department (e.g. quantities, product mix). In rough planning, generally done by the main contractor, a more detailed system concept considering the production equipment to be deployed is developed.

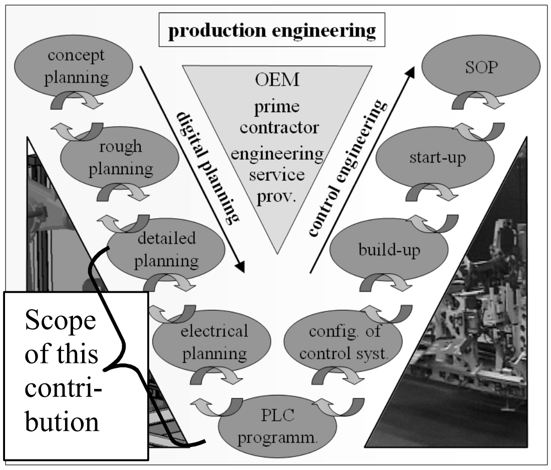

Figure 1: Key aspects of production engineering (Schmidgall, 2005)

In detailed planning, the specifics of each individual workstation must be worked out, including all the geometrical conditions (e.g. collision analyses, accessibility), capacity planning, time balancing, ergonomic analyses, sequences of operations, and off-line programming for automated components. With this detailed planning, a digital concept validation of the future system takes place by simulation (Strassburger, 2003). Afterwards the OEM grants approval for the subsequent realisation by the main contractor. The practical project work, i.e. PLC programming, control system configuration, build ups and the start-up phase for the overall system can be carried out.

Currently, the tools of digital planning cover exclusively the mechanics and robot planning and simulation. Integrated electric and logic control design of an entire cell in terms of digital planning has not yet been solved. This requires an interdisciplinary work between product design engineers, mechanic and manufacturer engineers, electric engineers as well as automation engineers. This paper focuses on the synthesis and simulation of the control applications and SCADA (Supervison Control and Data Acquisition) system configuration based on information of the digital planning.

2. MANUFACTURING CELL PLANNING

The planning process starts with the planning of the products. CAD tools are used which produce a file with all geometric data and product features. The next task designs the steps and phases of the production process. This is very essential because this functional approach designs the necessary logic of the production process first. The definition of the necessary resources implementing these procedures are defined after the functions are clear. There are some tools on the market which support this task by 3D graphic (animating the mounting activities in a 3 dimensional way) and semi-formal descriptions (e.g. Gantt Chart). Basic research work is done under the synonym of reconfigurable machine tools (Landers 2000) in connection with open software architectures (Pritschow 2001). These works aim to the optimisation of the kinematic of the changes in the product demands while this paper reports about generation of industrial PLC programs to minimize their programming and commissioning.

The three components of the digital planning product, process and resources have to be available as digital models. The product model is somehow standardised and implemented in the file exchange format for product data, e.g. in *.cgr files. For the scope of this paper the product information is not used.

There is no standard for the description of the process steps and phases. There are different implementations in different tools e.g. (Delmia, 2004) and (Tecnomatix, 2004). However it is essential that the purpose of a step and the signals which start the step and which indicate the end of a step are specified unambiguously. The following short example describes the process of a welding point where a clamp spans two product pieces, a robot sets a welding point and the clamp opens. The following steps with its synchronisation signals are the result of the detailed planing.

- Clamp is open
- Task: PutProductPiecesInClamp
- Sensor recognises that pieces are in the clamp: SenSigPiecesInCamp=1
- Shut the clamp: SigCloseClamp=1
- Task: ClampIsClosing
- Check if clamp is closed: SenSigClampClosed=1
- Task: SetWeldingPoint
- End of welding: RobotSigWeldingEnd=1
- Open the clamp: SigOpenClamp=1
- Task: ClampIsOpening
- Check if clamp is open: SenSigClampIsOpen=1
- Task: Remove the product piece

Here the written syntax is only used to show the involved tasks and signals. The concrete syntax is defined manufacturer and tool specific. There has to be a clear semantic behind the syntax to distinguish between tasks and signals as well as sequential and parallel processes. Additionally the signals have to be classified as signals which start tasks, signals which indicate the end of a task and signals which indicate events.

The necessary resources to implement the process steps are chosen as next task. The planning engineer usually knows which robot, clamp and other resources fulfil the requirements to implement the tasks. After this decision the tasks and the resources have to be linked together, i.e. the digital description of the resource and the description of the process step sequences are linked together. Figure 2 shows the planning results. On top of the figure it is shown a sequence of steps (shut and open a clamp and weld between these tasks). Beneath are shown the digital descriptions of the resources in our example of the clamp. The mechanic contains the CAD information which is not necessary for automation. The kinematic information is necessary for the simulation of the clamp. The description of the signals are necessary to export the start and indicate the end of the tasks. The electric information is needed for the E-CAD development process. It is out of the scope of this paper to deal with the representation of all this information. We can assume that it is possible to get the necessary information in a defined syntax and semantic as discribed in (Assmann, 2005).

If there is a functional description of the used resources, e.g. kinematic of a robot or of a clamp, then a simulation and animation is possible. The process description defines the order of the functions and the resource description carries out the functions of the equipment. Please recognise that we distinguish between task and function. The task describes the requirements of the production processes and the function describes the provided behaviour of the resources.

It is state-of-the-art to simulate and animate e.g. a robot cell based on the mentioned components. This simulation does not involve the behaviour of the controller which co-ordinates the resources. Usually the simulation environment generates and processes the synchronisation signals. It is also state-of-the-art that after fixing the detailed process steps the kinematic part of the robot program is generated.

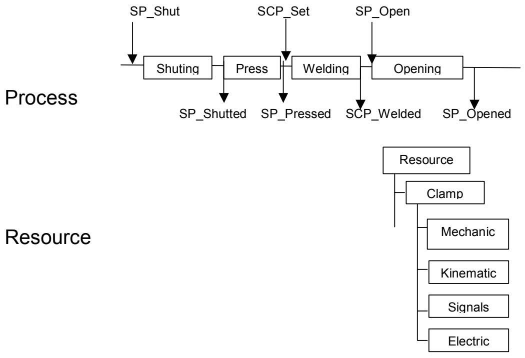

Figure 2: Planning results example (clamp and welding)

3. SIMULATION OF THE MANUFACTURING CELL

This paper reports about a concept to extend the simulation and animation of robot cells by controller and SCADA systems.

The intended simulation architecture is shown in Figure 3. It is assumed that the planning tools are able to simulate and animate the behaviour of the resources. This means that the motion of the resources, e.g. robot or clamp represents the exact motion of the real ones. Additionally the simulation environment of the planning tool provides interfaces which communicate the signals necessary to start the resource behaviour and indicate their positions as well specific events. Based on these signals the control program of the controller can control the resources. In our example the connection between the controller and the planning tool is implemented using OPC. As special extra of the robot cell simulation the SCADA system is conneted to the controller also. This is necessary to start and stop the production process. Additionally it is possible to simulate abnormal behaviour of the system as well as specific systems services e.g. for maintenance.

The connection between the controller and the SCADA system can also be implemented using OPC. Between the planning tool and the SCADA system there is no direct connection.

This concept is new because existing concepts usually simulate with full integrated controllers and visualisation. The authors do not know any simulation in terms of digital factories which involve SCADA systems. The great advantage is, that the kernel of the controller program and the SCADA configuration which are the basis of the implementation of the instrumented system can be validated.

One essential aspect of the concept is that the controller program and the SCADA configuration is generated out of the planning information. Starting point is the description of the process tasks and signals as well as of the resources as described above. This application design is the description of the behaviour of the controlled process. This means that it includes both the behaviour of the resources (e.g. kinematic of the robot or clamp) and the behaviour of the control application. Generating control applications means to separate both aspects. Only then the simulation of the resources in the planning tool and the execution of the generated control application can work together.

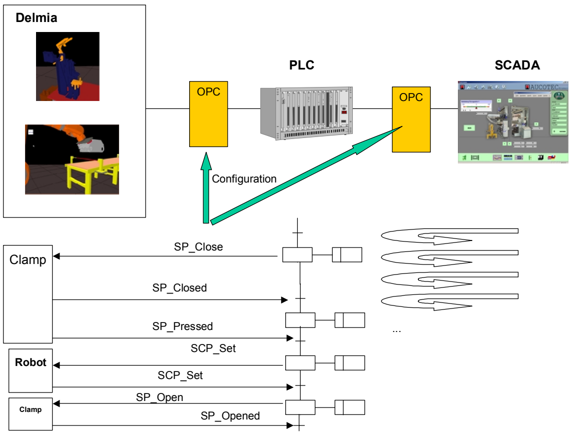

Figure 3: Robot cell simulation architecture including controller and SCADA systems

4. PLC PROGRAM GENERATION

4.1 General approach

The input for the control application generation comes from the result of the planning tool. The related information model is briefly described under clause 2 of this paper. It is a specific task to acquire the information out of the planning tool. The general concept is described in /Assmann 2005/.

The PLC program contains the logical control of the process sequences as well as the acquisition of sensor data and the control of actuators. Sensors and actuators are controller input and output signals which are called further on I/Os. The following mapping is defined based of IEC 61131-3 languages.

- The process sequences are transformed to the Sequential Function Chart (SFC) language
- The signals designed during the planning process become control program inputs and outputs (I/Os)
- The resources are transformed to function blocks which are carried out in the action blocks of the SFC.

The transformation of the process sequences to the SFC language is obviously. The integration of the instrumentation (field devices) in terms of pre-programmed function blocks is an extension to the simulation activities of digital planning tools. This provides the following new opportunities:

- Integrate control of cell infrastructure such as doors, emergency buttons
- In addition to the normal behaviour, the abnormal behaviour in terms of simulated errors (missing sensor signals, product faults) can be integrated in the simulation.
- Field device handling, e.g. commissioning can be part of the evaluation
- Operator actions can be simulated
- Some maintenance scenarios can be simulated additionally
- The controller program can reach nearly 80% of the final level.
- Performance evaluation of PLCs are done by nearly complete programs.

4.2 Rules for the transformation to SFC

It is obvious that the process tasks design in sequences and parallel tasks can be transferred to the IEC 61131-3 SFC language. This language consists of steps which carry out functions in so-called action blocks. The action blocks can generate events (e.g. binary variables) as a result of the action function which are used by other functions or transitions. Between these steps there are these transitions which are fired if an expression with a boolean result is true. One of the simplest expressions is a binary variable in our example SenSigPiecesInCamp=1. If the simulation is working without PLC in the planning tool, the signal which indicates the end of a task is used to start the following task (By the way this is the concept of IEC 61499, but this is not the scope of this paper). In a real robot cell or machine these signals are produced by sensors. For the transformation to the SFC the following mapping has been defined:

- Signals indicating the end of a task become controller inputs
- Signals starting tasks become controller outputs
- The controller inputs switch the transitions
- The controller outputs are the results of the action block functions or function blocks
- Sequential and parallel tasks are mapped one to one

4.3 Rules for the transformation to function blocks

The I/Os belong to resources, e.g. clamp and drives. Each of these resources consist of the “hardware”, e.g. motor, sensor, clamp AND the transformer form/to the physical signal(s) (e.g. rotation, way, pressure, force) to/from the electrical/pneumatic/ hydraulic and at least the digital signal(s). These transformers are known as field devices. These field devices are controlled by more then one signal. Therefore the I/Os are part of function blocks representing the resources. The function blocks are usually composite function blocks i.e. a resource function block contains more than one function block which interact with each other. For example a function block for a turn table consists of the function block controlling the drive, sensors to indicate the position of the table as well as sensors to indicate abnormal situations, e.g. torque and temperature measurement. Some field devices deliver signals only (e.g. position sensors) while some are more complex and need a specific function block for program integration (e.g. torque measurement). The composite function block contains all these function blocks as well as additional functions and function blocks to provide the behaviour of the turn around table at the interface.

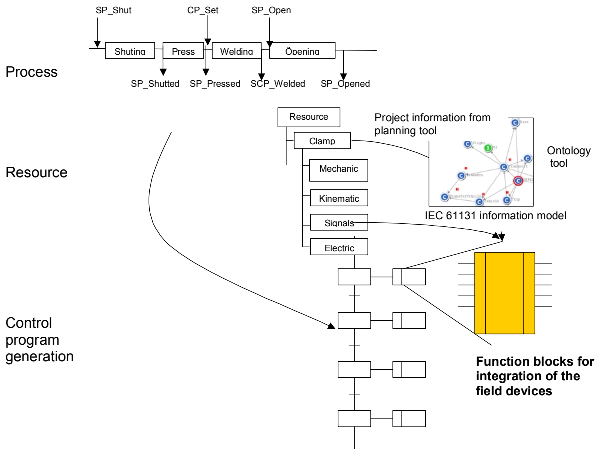

Figure 4: Transformation of designed process task to IEC 61131-3 SFC and function blocks

4.4 Transfer syntax

The standard IEC 61131-3 defines the languages in terms of syntax and partly semantic. This syntax is related to the source code text which is edited with the so-called integrated development environment (IDE) of the controller. The syntax of the textual languages such as structured text and instruction list can be used partly as transfer syntax. The graphical languages such as SFC, function block diagram or ladder diagram have no textual representation in the standard. An XML based transfer syntax for SFC was defined by the industrial association PLCopen. This organisation supports the standard in terms of marketing and technical additions. In the scope of the project Modale, this XML syntax is used to describe the SFC. The IDE of the used PLC has to provide a related import filter.

The function blocks carried out on the action blocks of the SFC are instances of function block classes of an IDE library, in the project Modale the openPCS tool by Infoteam (2004). The function blocks are programmed and tested in advance for each resource, i.e. the algorithms which interact with the field devices are not generated, they are pre-programmed. The selection is done by name, i.e. during planning the resources and the tasks are related to each other.

The described transformation is done under the support of the ontology tool KAON by FZI (2005). For more details please see Assmann (2005)

5. SCADA CONFIGURATION GENERATION

The SCADA system provides to the worker an overview about the current situation and gives them opportunities to control the general process (e.g. start, stop). Additionally the SCADA systems provide services for diagnosis and maintenance. The SCADA system has screens which show the structure of the machine or robot cell where the components show data about their current state. The configuration of the SCADA system has to design the screens and to relate information in terms of data to the components. Configuration means collect all symbols and relate data to the attributes of the symbols. The SCADA is represented by a information expressed in a UML class diagram and exchanged using XML files. The SCADA model, in the Modale project AucoVision (2005) consists mainly of (Figure 5) SymbolList, PictureList, GroupList, InstanceList, JobList and ItemList (Alarm, SMS, Maintenace).

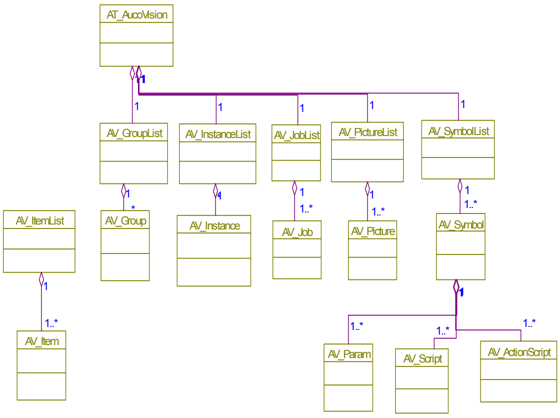

Figure 5: Class diagram of the SCADA system

The digital planning has defined the resources which are represented in the planning results as shown in Figure 2. Similar to the approach for the integration of the function blocks in the SFC, the symbols are pre-defined and available in a SCADA library. Each symbol has a defined structure of attributes which has to be linked to data. The data comes from the controller and are the function block inputs and outputs as well as some additional signals. In other words the configuration needs the same structural information as the controller. The SCADA system needs additional inputs from the digital planning:

- Resource names to select the symbols
- Resource features e.g. MTBF data or maximum number of switches which are mapped to diagnosis and maintenance procedure of the SCADA system
- Operator interactions controlling the process e.g. emergency stop, manual intervention which interact with the PLC program

After the import of the generated SCADA project there are three mechanisms for the configuration:

- manual position from import list including all attributes
- import list contains position of symbols and SCADA system generates the figures
- predefined frame which is configured by the instances of the import list

In first mechanism the configuration engineer of the SCADA system has to place the full list of symbols manually. This gives the developer good support but this mechanism is only partly automated.

The second mechanism is the high end because all configuration activities can be fully automated. In principle the arrangement of the machine or robot cell defined during the planning process can be used. Up to know there is no standard to export this information from the planning tool.

The third mechanism is similar to a pin wall. The pin wall is the screen. Usually the screen has a fixed frame e.g. on top alarms and urgent information, in the middle the symbols and at the bottom buttons for interaction with the system. Now the symbols which have a specific order in the transfer file are selected and placed at the right position of the screen. As mentioned earlier the inputs and outputs of the function blocks become the attribute.

6. SUMMARY

The paper describes the integration procedure of controller and SCADA systems into digital planning and simulation. This integration is an extension to the current situation where the simulation is organised within the planning tools and the generation of programs is restricted to robots. The generation of controller programs and SCADA system configuration are described in detail.

The approach is implemented in a demonstrator software consisting of an emulation of resource behaviour, a task sequence editor and a generator of the controller programs and SCADA system configuration. The demonstrator works with the tool openPCS by infoteam and AUCOVision by Aucotec.

ACKNOWLEDGEMENT

The development of some parts of the concepts mentioned in this article are supported through the publicly funded BMBF project (Deutsches Bundesministerium für Bildung und Forschung) MODALE (Modellbasiertes Anlagenengineering, kundenorientierte Dienstleistungen für Anlagensteuerung und –kontrolle). More information about this project is available under Modale (2004).

REFERENCES

Delmia (2004). Homepage. *URL: http://www.delmia.de*

Assmann, D., et. al. (2005). Using Ontology-Based Reference Models in Digital Production Engineering Integration. *Submitted to the 16th IFAC World Congress, Prague, Czeck Republic.*

Landers, R.G., 2000: „A New Paradigm in Machine Tools: Reconfigurable Machine Tools". *Japan-USA Symposium of Flexible Automation, Ann Arbor, Mchigan, July 23-26.*

IEC 61131-3 (1992): Programming Language for Programmable Controllers, *IEC Geneva 1992.*

Pritschow, G. et.al. : "Open Architecture Controller ", *Annals of the CIRP*, Vol. 50/2.

Schmidgall, G. Kiefer, J., Bär, T. (2005): Objectives of Integrated Digital Production Engineering in the Automotove Industry. *Submitted to the 16th IFAC World Congress, Prague, Czeck Republic.*

Strassburger, S., Schmidgall, G. , Haasis, S. (2003). Distributed manufacturing simulation as an enabling technology for the digital factory. *Journal of Advanced Manufacturing Systems*, **Vol. 2, No. 1**, 111-126

Tecnomatix (2004). Homepage. *URL: http://www.tecnomatix.com*