INTERNATIONAL CONFERENCE ON ENGINEERING DESIGN
ICED 03 STOCKHOLM, AUGUST 19-21, 2003

CONCEPTUAL DESIGN APPROACH FOR MECHATRONIC SYSTEMS CONTROLLED BY A PROGRAMMABLE LOGIC CONTROLLER (PLC)

J Bathelt, A Jönsson, C Bacs, A M Kunz and M Meier

Abstract

Traditionally, the development of mechatronic products starts with the mechanical design. Later electronic components and program logic for the product are added. This work proposes an improved interdisciplinary concurrent product development process with an extended function structure, starting from the conceptual design phase.

The new methodology is based on the German design guideline VDI 2221 [1]. In this work transition conditions, time aspects and logic flow are added to the normal function structure. An example shows how to use the extended function structure.

The extended function structure is used to derive the initial logic used for the development of the PLC program controlling the machine later on. Since the extended function structure also covers the traditional information, it can still be used when building the assembly tree.

By adding more information in the neutral function structure, the same information can be used for the mechanical design as well as for the control logic. The interdisciplinary communication and documentation among the engineers will be improved and errors in the concept will be detected earlier.

Keywords: Electrical engineering, Introduction of methods in industry, Mechanical product design, Mechatronics.

1. Introduction

The generally applicable magic triangle [2], consisting of shortening time, improving quality and reducing costs, is an ongoing challenge for interdisciplinary teams developing mechatronic systems. The aim of this work is to improve the concurrent engineering of machines controlled by a PLC, in order to find and avoid errors early and to face the time-to-market demand. PLC based mechatronic systems are common and typically used on the machine level, including several actuators and sensors [3].

Today, the mechanical engineer usually leads the development process of mechatronic systems as shown in the upper part of Figure 1. In many cases, the implementation of the control logic starts after most of the CAD data is generated, or sometimes even after the hardware is manufactured. The interdisciplinary data exchange is not standardised. It is obvious that interdisciplinary errors are detected rather late in those development processes, where they have a high impact on the overall product development costs [4], [5].

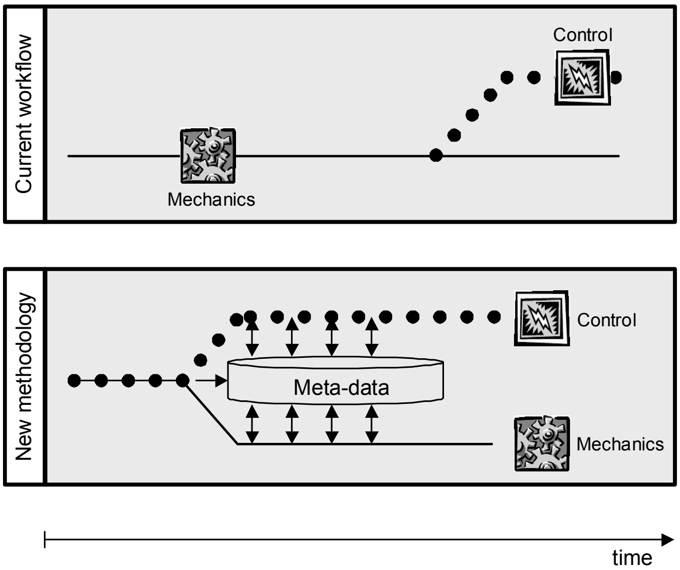

Figure 1. Current and proposed workflow.

The workflow within the proposed methodology will include both disciplines from the beginning as shown in Figure 1. In order to cover all information needed for the concept and embodiment phase [1], [6] an extended function structure will be presented in this work.

2. Background

The function structure, described in [1], [4], [6-8], is widely accepted as a bridge between analysis and synthesis [9]. It is describing the overall functional behaviour supported by the concepts of hierarchy and sequence. Three flows: material, energy and signal can be assigned to the sequence in order to specify the in- and outputs of the functions. In this work a fourth flow: *logic*, is introduced as shown in the principle layout of a mechatronic system controlled by a PLC, see Figure 2.

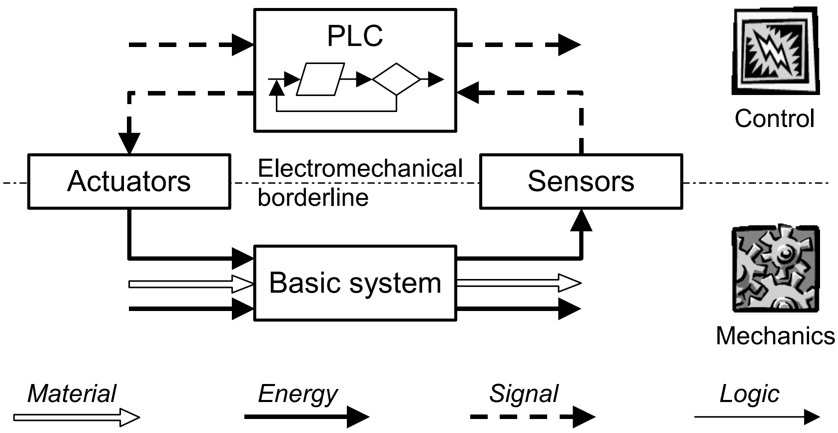

Figure 2. Schematic layout of a mechatronic system controlled by a PLC.

In this case the signal flow is exchanging information between the actuators/sensors and the control via wires. The logic flow represents the control logic, typically implemented by using one of the five languages for PLC's specified in IEC 1131-3 [10]. The two different flows are used above the electromechanical borderline in order to be able to distinguish between the control logic, and the signals between the actuators/sensors and the control.

Similar figures for general mechatronic systems (without the fourth logic flow) can be found in the new VDI guideline for the development of mechatronic systems [11], or in [12], [13]. To keep the picture simple, the necessary power supply for the actuators, the sensors and the PLC as shown in [12] are not drawn. According to the electromechanical borderline shown in Figure 2, the control engineers are mostly interested in the logic and signal flow and the mechanical engineers in the energy and material flows.

3. Proposed development process

The overall layout for the proposed interdisciplinary design process is shown in Figure 3 and is based on standards and guidelines [1], [6], [7], [11].

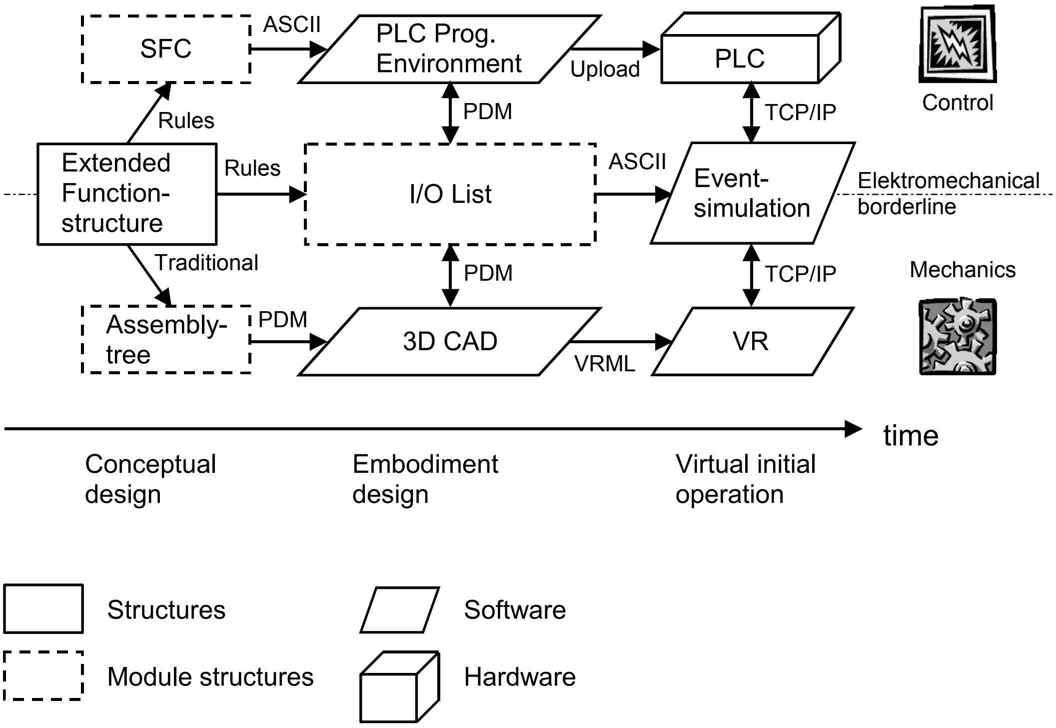

Figure 3. Proposed development process.

Today, it is common to formulate a rough assembly tree inside a PDM system. The assembly tree is then used to initiate a concurrent workflow for the embodiment of CAD geometry.

A similar mechanism exists for the development of PLC code. The graphical programming language SFC (Sequential Function Chart) is intended to initiate the programming of the PLC [10]. SFC is one out of five standardised programming languages specified in IEC1131-3. It can be mixed or translated to the other languages during the coding phase.

In this work a new extended function structure is presented in order to be able to derive the assembly tree, SFC and the I/O list as shown in [14]. The I/O list is a flat list containing a description of all sensors (inputs) and actuators (outputs). Since the actuators and sensors are located on the electromechanical borderline (Figure 2), this widespread used list is helpful when establishing a bridge between mechanics and control.

The bi-directional event-based connection between the PLC and the geometry generated using CAD, visualised in a virtual reality environment, allows the virtual initial operation (Figure 3). The concept of a virtual machine is described in [15] and is also commercialised [16].

4. The extended function structure

The need of an interdisciplinary graphic design language for the early phases of the development of mechatronic systems is stated in [17]. It has also been found that transition conditions and the time aspect are important issues for mechatronics [18]. The importance of transition conditions for mechatronic systems controlled by a PLC is even stronger [19]. That's because a PLC implies sequential behaviour including time-delayed steps and (transition) conditions before the next step may occur.

The extended function structure also contains the logic flow introduced in Chapter 2. This will help to derive and distinguish the flows. The look and feel of the proposed extended function structure is a mix of existent solutions and SFC. For instance, the hierarchy is symbolised by drawing sub-functions inside of their main functions as in SFC, FAST (Function Analysis System Technique) or SADT (Structured Analysis and Design Technique). Transition conditions could be drawn as a function in a box. Since they often imply a sensor, they are drawn as a perpendicular borderline, which has to be passed after the condition is fulfilled, like in SFC or Petri nets.

The main aspects of the current function structures are:

- Hierarchy
- Sequence
- Material-, Energy- and Signal flow

The extensions to optimise the development process of mechatronic systems controlled by a PLC are:

- Transition conditions
- Time aspect
- Logic flow

The usage and the appearance of this extended function structure is shown in the following chapter, using an automatic door opener as an example.

5. Example

Since automatic door openers for humans are common worldwide, the example is easily followed. The main function for this mechatronic application is shown in Figure 4. The four flows material, energy, signal and logic will be separated using the conventions shown in Figure 2.

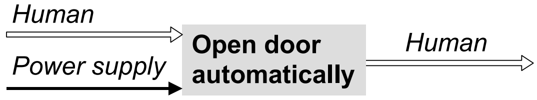

Figure 4. Main function for an automatic door opener.

The first hierarchy level could look like the second function structure shown in Figure 5. In order to open the door automatically, a functionality to detect a person in front of the door is needed. This is drawn as a transition condition in Figure 5. The time aspect is explicitly shown in the “Wait 9 seconds” function. A guessed or specified time-delay could be added to all functions as an attribute.

If a human is detected while the door is closing, it should open immediately. This is taken into account through the decision between the two transition conditions ‘Human detected’ and ‘Door closed’. Like in Petri nets, one of these conditions must be fulfilled to proceed.

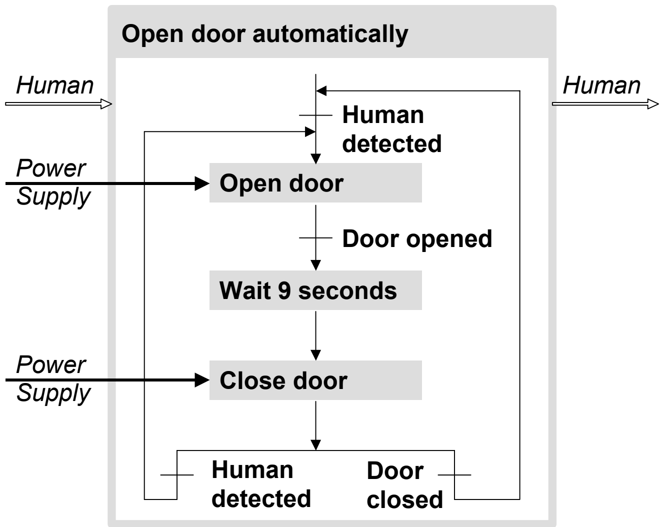

Figure 5. Second function structure of an automatic door opener.

The next hierarchy level is introduced in Figure 6. Typical functions like ‘Generate opening force and -motion’ are now visible. This function has two inputs (Opening signal and Power supply) and the mechanical energy as an output. As visualised in Figure 2 and shown in [14], this function leads to an actuator.

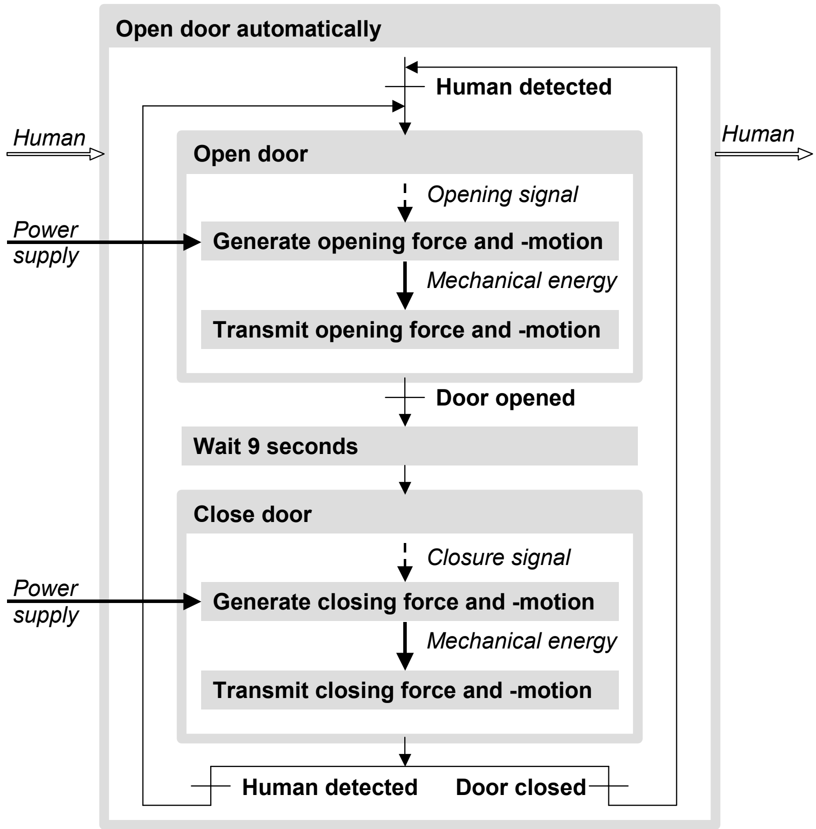

Figure 6. Third function structure of an automatic door opener.

Transition conditions are leading to sensors [14]. For instance the functionality 'Human detected' in Figure 5 or Figure 6, will be implemented by the use of some kind of sensor. The information about the variable size is already available at this stage. Since this sensor has to send a binary signal, it is a so-called 'digital input'. A variable name can also be set. The summarized sensor information is stored in the first line of Table 1. Similar data can be derived for the door sensor and actuator.

Table 1. I/O list for automatic door opener.

Id	Type	Size	Variable	Actuator/Sensor Description	Comment
1	Input	Digital	`di_hum_detected`	Human detector	
2	Output	Digital	`do_open_door`	Door actuator	Generate translation force and -motion
3	Output	Digital	`do_close_door`		
4	Input	Digital	`di_door_opened`	Door sensor	
5	Input	Digital	`di_door_closed`		

Pure mechanical functions, like 'Transmit opening force and -motion', are suppressed in order to derive the control logic [14]. The main control structure is expressed in SFC and shown in Figure 7. As shown in the 'Close door' function, it is possible to refine the control logic. In this case the door will only be closed, if no human is detected.

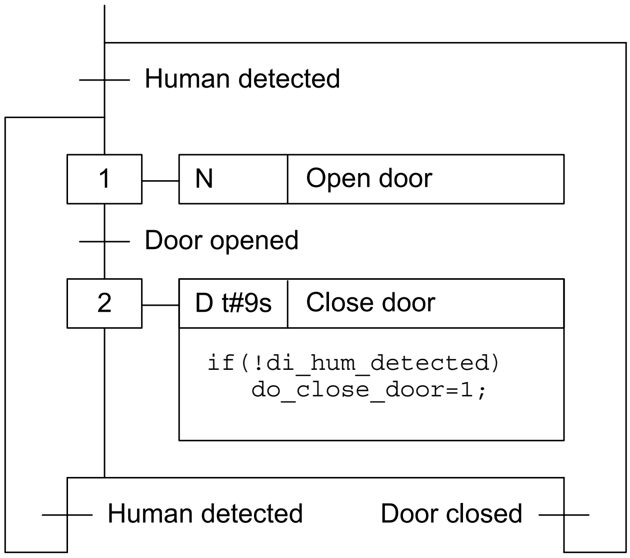

Figure 7. SFC derived from the third function structure.

The *intra*disciplinary modules ‘SFC’ and ‘assembly tree’ are now used to start the embodiment design phase as shown in Figure 3. A more complex example, regarding the numbers of actuators and sensors, is the combing machine from textile industry presented in [14].

6. Conclusions and future work

The need to improve the product development process for mechatronic systems is obvious. The proposed enhanced product development process for PLC-controlled mechatronic systems is based on an extension of the traditional function structure. The extended function structure also includes transition conditions, time aspects and logic flow. The extended function structure is the fundament for a short path to SFC, assembly tree and I/O-list in order to initialise the embodiment design phase using a 3D-CAD and a PLC programming environment.

In order to include control logic issues in the extended function structure, a multidisciplinary team with broad knowledge should build the extended function structure. This allows enhancing the conceptual design phase, due to a more creative working environment.

When extending the function structure, the traditionally layout has to be modified to include the new features. However, this modification is small, and the user is still familiar with the new function structure layout.

Work has been done and published to describe and improve the conceptual design phase for the development of mechatronic products in general. No work has been found where the implied restrictions of a PLC are used to optimise the conceptual design phase.

The modularisation resulting from the extended function structure, presented in this work, and the use of the rules, presented in [14], is developed in order to improve the workflow in the interdisciplinary team. During the embodiment phase the team members have to use their specific tools (Figure 3), which leads to a “modularisation by separation”. The necessary interdisciplinary bridge is established by the use of an I/O list and a PDM system. Other studies are usually focussing on the product structure and the configuration of mechatronic products, which leads to a ”modularisation by integration” approach [20], [21], [22].

The information flow and documentation in the product development process will be improved using the general information covered by the extended function structure, as well as by the PDM connection between the CAD system, the PLC programming environment, and the I/O-list. Furthermore the virtual prototyping, by the integration of the Virtual Machine [15] into the proposed development process as shown in Figure 3, will be prepared. Since the development of the program for the PLC can start earlier and an interdisciplinary bridge is established, errors will be detected earlier, costs will be reduced, and the development time will be shortened.

In the future, the following topics will be addressed:

- Development of Software to handle the extended function structure, the SFC, the I/O list and the assembly tree and the PDM connections between the PLC programming environment, the CAD system and the Virtual Machine.
- Continuing testing the concept in industry and finally also testing the software in industry.

References

[1] "Systematic Approach to the Design of Technical Systems and Products", VDI Design Handbook 2221, VDI-Verlag, 1987.

[2] Bullinger H-J., Ohlhausen P., Prieto J. and Bröcker A., "Concurrent simultaneous Engineering (CSE) – New orientation in product development", Proceedings 1st CME-Congress, Bremen, 1998.

[3] Aarnio J. P. and Riitahuhta A. O., "Modularization of the control system in a mechatronic product", Proceedings of ICED '01, Vol. 3, Glasgow, 2001, pp.555-562.

[4] Ehrlenspiel K., "Integrierte Produktentwicklung", Carl Hanser Verlag, München, 1995.

[5] Lückel J., Koch Th. and Schmitz J., "Mechatronik als integrative Basis für innovative Produkte", VDI Berichte Nr. 1533, VDI Verlag, Düsseldorf, 2000.

[6] Pahl G. and Beitz W., "Engineering design: a systematic approach", Springer, Berlin, 1999.

[7] Hubka V. and Eder W. E., "Einführung in die Konstruktionswissenschaft", Springer, Berlin, 1992.

[8] Breiing A. and Flemming M., "Theorie und Methoden des Konstruierens", Springer, Berlin, 1993.

[9] Pulm U. and Lindemann U., "Enhanced systematics for functional product structuring", Proceedings of ICED '01, Vol. 1, Glasgow, 2001, pp.477-484.

[10] Bonfatti F., Monari P. D. and Sampieri U., "IEC 1131-3 Programming Methodology", CJ International, France, 1997.

[11] "Entwicklungsmethodik für mechatronische Systeme", VDI Design Handbook 2206, VDI-Verlag, Düsseldorf, 'Gründruck' 2003.

[12] Gausemeier J. and Lückel J., "Entwicklungsumgebungen Mechatronik-Methoden und Werkzeuge zur Entwicklung mechatronsicher Systeme", Heinz Nixdorf Institut-Verlagsschriftenreihe, Bd. 80, 2000.

[13] Kallmeyer, F., "Eine Methode zur Modellierung prinzipieller Lösungen mechatronischer Systeme", HNI-Verlagsschriftenreihe, Paderborn, 1998.

[14] Bathelt J., Jönsson A., Bacs C. and Meier M., "Modularisierung SPS-gesteuerter mechatronischer Systeme", Proceedings of the 1st Paderborner Workshop: Intelligente mechatronische Systeme, 2003.

[15] Dierssen S., "Systemkopplung zur komponenten-orientierten Simulation digitaler Produkte", Fortschr.-Ber., VDI Reihe 20 Rechnerunterstützte Verfahren, Nr. 358, VDI-Verlag, Düsseldorf, 2002.

[16] Intelliact AG, http://www.intelliact.ch/, 2003.

[17] Buur J. and Andreasen M. M., "Design models in mechatronic product development", Journal of Design Studies, Vol. 10, Nr. 3, 1989, pp.155-162.

[18] Buur J., "A Theoretical Approach to Mechatronics Design", Technical University of Denmark, Dissertation, 1990.

[19] Santos A. P., Cury J. E. R., De Negri V. J., "Mechatronic Design and Supervisory Control Theory for Fluid Power Applications", 5th International Conference on Fluid Power Transmission and Control, Hangzhou, China, 2001.

[20] Aarnio J. and Riitahuhta A., „Modularisation by integration", Proceedings NordDesign 2002, Trondheim, 2002, pp.1-8.

[21] Svensson D., Hallin K., Zimmerman T. and Malmqvist J., "An information model for mechatronic products focusing on early development phases", 6th Workshop on Product Structuring – application of product models, Copenhagen, 2003.

[22] Möhringer S. and Gausemeier J., "An interface specification for principle solutions supporting the cross domain design of mechatronic systems", International design conference – Design 2002, Dubrovnik, 2002, pp.533-538.

Jens Bathelt, Christian Bacs, Andreas M Kunz, Prof. Dr. Markus Meier
Federal Institute of Technology, Institute of Mechanics, CH-8092 Zurich, Switzerland
Tel: +41 1 632 35 15, Fax: +41 1 632 11 81, E-mail: bathelt@imes.mavt.ethz.ch
URL: http://www.zpeportal.ethz.ch

Anders Jönsson
Blekinge Institute of Technology, Department of Mechanical Engineering, SE-371 79 Karlskrona, Sweden
Tel: +46 455 38 55 12, Fax: +46 455 38 55 07, E-mail: anders.jonsson@bth.se
URL: http://www.bth.se/ima