LUND UNIVERSITY

Towards a seamless integration between process modeling descriptions at Business and Production levels - work in progress

Gerber, Tobias; Theorin, Alfred; Johnsson, Charlotta

Link to publication

Towards a seamless integration between process modeling descriptions at Business and Production levels - work in progress

Tobias Gerber*, Alfred Theorin**, Charlotta Johnsson***

* German Research Center for Artificial Intelligence,
Trippstadter Strasse 122, 67663 Kaiserslautern Germany,
(email: tobias.gerber@dfki.de)
** Department of Automatic Control, Lund University, Sweden,
(e-mail: alfred.theorin@control.lth.se)
*** Department of Automatic Control, Lund University, Sweden,
(e-mail: charlotta.johnsson@control.lth.se)

Abstract: To fulfill increasing requirements in the manufacturing sector, companies are faced with several challenges. Three major challenges have been identified regarding time-to-market, vertical feedback loops and level of automation. Grafchart, a graphical language aimed for supervisory control applications, can be used from the process-planning phase, through the implementation phase and all the way to the phase for execution of the process control logics, on the lower levels of the Automation triangle. Work in progress is examining if the same concepts could be used on the higher levels of the Automation triangle as well. By splitting the execution engine and the visualization engine of Grafchart various different visualization tools could potentially be used, however connected by the shared Grafchart semantics. Traditional Business languages, e.g. BPMN, could therefore continue to be used for the process-planning phase whereas traditional production languages, e.g. Grafchart or other SFC-like languages, could be used for the execution. Since they are connected through the semantics, advantages regarding the three identified challenges could be achieved; time-to-market could be reduced, the time delays in the vertical feedback loops could be reduced by allowing KPI visualization, and the level of automation could be increased.

Keywords: Enterprise-wide Information System, Manufacturing System Engineering; Business Process Modeling.

1. INTRODUCTION

To fulfill increasing requirements in the manufacturing sector, companies are faced with several challenges. Three major challenges have been identified;

1) The first challenge concerns the time-to-market, i.e. the amount of time needed to go from setting of the objective to the planning phase, through the commissioning phase to the operation phase (time-on-market), see Figure 1 (Kiefer, 2007). The time-to-market needs to be as short as possible.

2) The second challenge concerns the vertical feedback loops, i.e., the possibility to bring production data up to management level and bringing management decisions down to the production level. The time delay in the vertical feedback loop should be as short as possible. This is sometimes expressed as "closing the gap between enterprise resource planning" (AberdeenGroup 2006) or "poor visibility into plant operations" (Forrester, 2000).

3) The third challenge concerns the level of automation, i.e. the level of the business' functional hierarchy that can be automated. The level of automation should be as high as possible (Johnsson, 2008).

This paper aims at presenting some conceptual ideas of how these three challenges can be met.
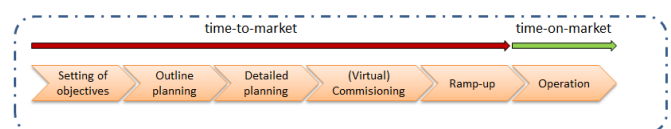

Figure 1: Illustration of the time periods time-to-market and time-on-market.

2. BASIC CONCEPTS

Important theories and concepts that are needed when working with these three challenges are; Collaborative

manufacturing model (Gorbach & Nick, 2002), functional hierarchies (IEC 62264-1, 2000), control architecture (Johnsson, 2008) and performance metrics (draft ISO 22400-Part1, 2011). The problem has also been worked on in earlier research projects such as e.g. CIMOSA (CIMOSA).

2.1 Collaborative Manufacturing Model

The collaborative manufacturing model, see Figure 2 (Gorbach & Nick, 2002), reflects the fact that companies today need to collaborate both externally and internally to reach success on the market.

The external collaboration is presented through the two horizontal axes. The value chain axis highlights the need for cooperation with suppliers and customers. The lifecycle axis goes from design to support of a product.

The vertical axis, i.e. the enterprise axis, represents the need for internal cooperation from the business side to the production side, sometimes expressed in statements such as "shop floor to top floor", "sensor to boardroom" or "plant to enterprise".
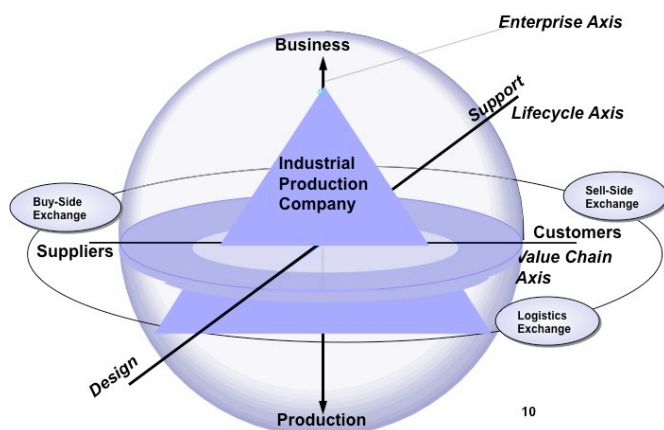

Figure 2: A variant of the Collaborative Manufacturing Model developed by ARC (Gorbach & Nick, 2002)

Model Based Systems Engineering can be compared with the Lifecycle axis in the Collaborative Manufacturing Model. Model Based Systems Engineering is considered as a formalized application of modeling to support system requirement, design, verification and validation, beginning in the conceptual design phase and continuing throughout development and later life cycle phases (INCOSE 2007). Perpendicular to the Lifecycle view is the Enterprise view, i.e., the view of transforming generic business strategies into detailed production execution tasks.

This article focuses on the intersection of the Lifecycle axis and the Enterprise axis. Only a reduced section of the Lifecycle axis is considered, namely the development, commissioning and execution phase. By being able to use the same tool, for example the same graphical language, in both the design and execution phase, the time-to-market can be reduced.

2.2 Functional Hierarchies

The functionalities needed for controlling a company internally are hierarchically structured in a functional model (IEC 62242, 2003), see Figure 3. Figure 3 depicts the different levels of the functional hierarchy model: business planning and logistics (level 4), manufacturing operations and control (level 3) and batch, continuous, or discrete control (level 1-2). The levels provide different functions and work in different timeframes.

- Level 4 defines the business-related activities needed to manage a manufacturing organization. Manufacturing-related activities include establishing the basic plant schedule (such as material use, delivery and shipping), determining inventory levels and making sure that materials are delivered on time to the right place for production. Level 4 typically operates on time frames of months, weeks and days.

- Level 3 defines the activities of the work flow to produce the desired end-products. It includes the activities of maintaining records and coordinating the processes. Level 3 typically operates on time frames of days, shifts, hours,

- Level 2 defines the activities of monitoring and controlling the physical processes. Level 2 typically operates on time frames of hours, minutes, seconds and sub-seconds.

- Level 1 defines the activities involved in sensing and manipulating the physical processes. Level 1 typically operates on time frames of seconds and faster.
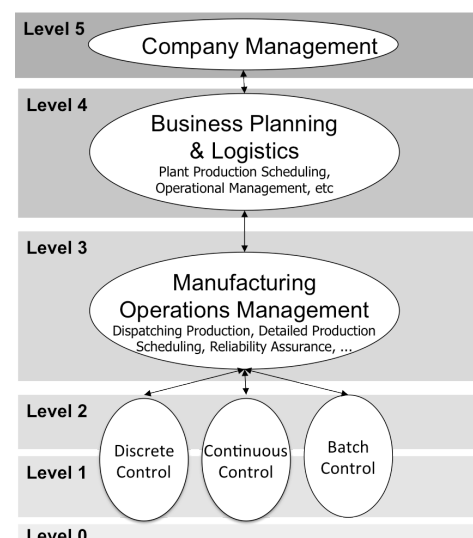
- Level 0 defines the actual physical processes.

Figure 3: Functional hierarchy (IEC 62264, 2003)

Figure 3 shows that the lowest level of control, i.e. control at level 1-2, differs with respect to the type of production (batch, discrete and continuous), whereas control at the upper levels, i.e. level 3 and 4, is more similar.

This article focuses mainly in the integration from level 4 to level 3-2. The integration from level 3-2 to level 1-0 is covered in other articles such as (Theorin et al, 2012). By being able to integrate level 4 to level 3, it will be possible to extend the vertical feedback loop. Data from the production plant (level 0-1) that are captured and stored in historical databases (level 2) can be transformed into performance metrics (level 3) and sent to the Business planning and logistics (level4).

2.3. Control Architecture

Today, control architectures consist of several layers that realize different automation tasks (IEC 62264, 2003), see Figure 4. The control level closest to the physical plant, referred to as level 1,2 in IEC 62264, is taking care of specific control tasks. This code is most often implemented in Programmable Logic Controllers (PLCs). The specific control tasks need to be coordinated and orchestrated. This is taken care of by the supervisory control layer and the MES control layer (referred to as level 2 and 3 respectively in IEC 62264). The complete production process is taken care of by the business control layer (level 4 in IEC 62264).
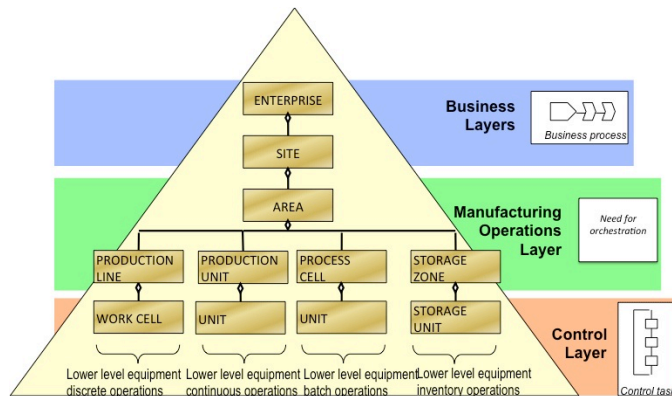

Figure 4: Control Architecture

Industry standard today is to implement the control tasks, i.e. program the PLCs, according to the international standard IEC 61131-3 whereas natural language or simple sketches are used for the specification of the control tasks. The specification of the complete production process is frequently done using business processes, e.g. BPML, whereas an industry standard for implementation and execution at this level still is undefined. The role of the supervisory control layer and the MES layer is to coordinate and orchestrate the control tasks but also to be the link between the business processes and the control tasks. There is currently a large interest in the supervisory control layer and the MES layer.

This article focuses mainly on the integration of BPMN descriptions traditionally used for modeling and visualization of business processes and MES procedure descriptions who's aim is to coordinate, execute and control the automation tasks in the levels below. By bringing the BPMN process description closer to the MES procedure descriptions the level of automation can be increased.

2.4 Performance Indicators

A key performance indicator shall be one or several related measurements that provide a proxy understanding of the overall progress of the business for those who use it. A good KPI must be specific, actionable, sensitive to change, reliably measurable and well-understood by those who use it (draft-ISO 22400-Part1).

The KPIs are based on data from the production plant (level 0-1), the data could be captured on line or retrieved from a historical database (level 2). The data is being transformed or aggregated into more elaborated parameters (level 3). Some of the KPIs are of usage by level 3 operators, e.g. production line managers, whereas others are further transformed or aggregated to KPIs that are of relevance for level 4. More information about KPIs are found in (MESA 2006) and (Draft-ISO 22400-Part2).

Examples of KPIs that are of relevance for level 3 (draft ISO 22400-2, 2011): OEE for single machines, availability of single machines, and allocation degree for single machines.

Examples of KPIs that are of relevance for level 4 (MESA, 2006): Labour cost per unit, manufacturing cycle time, and on time delivery.

3. GRAPHICAL LANGUAGES

The usage of graphical languages has many advantages such as giving the user a good visual interpretation of what is happening (compare "a picture says more than a thousand words"), the user often thinks it is easier to program by graphical means than by textual means, increased possibility to use colours, etc. In this spirit, the possibility to have a common, or similar graphical language is very appealing. A common language could:

- be used for design and execution (compare challenge 1),

- include support for performance metrics visualization (compare challenge 2) and

- could be used at all functional and control levels seems very appealing (compare challenge 3).

A well developed graphical language should have support for formal analysis, have a well defined and limited number of graphical elements, good abstraction facilities, and a solid base in its syntax, semantics, and pragmatics (Johnsson, 2008).

3.1 Situation at present

The situation concerning the usage of graphical languages could be summarized in the following way (Johnsson, 2008):

- Level 1-2 (Process control systems): Standardized graphical languages are used for basic control issues e.g. PLC programming. One popular language is SFC (reference IEC 61131-3).
- Level 2-3 (Supervisory control and MES): Standardized graphical languages exist and are, together with some proprietary graphical control languages, used for supervisory control issues. None

or only local graphical control languages are used for MES issues.

- At the levels 1-3 (PCS, supervisory control and MES level) the languages are used for executing control over the production process. The graphical language has a bi-directional communication with the plant, i.e. data can be collected from the plant and visualized for the user by the means of a graphical language, or data can be sent to the plant through the graphical language.
- The syntax and semantics of the languages used at the PCS, supervisory and MES level are very similar.
- Level 4: Local graphical languages are used for visualising the business processes. Initiatives exist striving to standardize the language used for visualization of business processes.
- At level 4, the languages are used for presenting the business processes. The languages do not have a communication with the production plant.
- The syntax of the languages used at level 4 is different from the one used at the PCS, supervisory and MES level.

3.3. Syntax vs. Semantics

A process language is formally defined by its syntax and its semantics. The syntax is the notation being used and the semantics is the interpretation of the symbols.

Within enterprise applications, process-oriented management and development are already state of the art to enable a high flexibility for workflows such as marketing, sales, and purchasing. Hence, on business level there exist established methods for process modelling, e.g. BPMN (OMG, 2010), most of them have their focus on the syntax rather than on the semantics. However, no standardized process modeling language exists for the technical view on the production process, see Figure 5. Proposals exists, e.g., various SFC-like languages, most of them have both syntax and semantics.
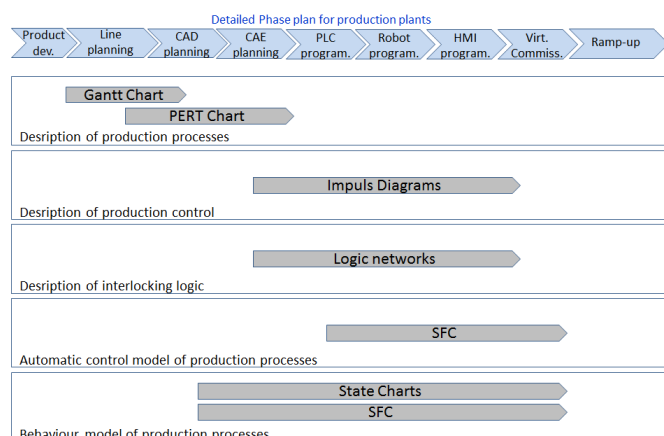

Figure 2: Modeling Languages during the production process planning (Draht, 2010)

A process is the totality of mutually influencing operations within a defined arrangement of components which are in relation to each other by which material, energy, or information is transformed, transported, or stored (DIN 19226-1). Generally, manufacturing processes define an arrangement of workflows which are executed by technical equipment or humans. Compared with the business process there is no difference except that one perspective on the process is technical and one is business oriented.

Modeling the manufacturing processes, both from a high-level business process level and from a technically executable process level, could therefore be done with the same methods and modeling techniques. This engineering approach is known as process-oriented planning (Pohlmann, 2008). Since the usage of the same formal representations increases the transparency, the approach of an integrated process modeling helps closing the information gap between business and technical manufacturing levels.

The reason why different modelling languages are used at different levels within a company is mainly the specialised focus of the users. Today fast working processes in all levels of manufacturing companies is needed to face the well-known drives e.g. shorten product lifecycles or customer individuality. Therefore every process modelling language has its right to exist. But to fasten up the seamless process integration over all levels of hierarchy and within all levels of planning phases a strict detachment of the syntax and the semantic is needed.

4. GRAFCHART

4.1 Basic ideas

Grafchart is a graphical programming language for sequential, procedural, and state-transition oriented applications (Johnsson, 1999). Grafchart has been developed at Lund University, Sweden since 1991 (Arzén, 1991). It is based on ideas from Sequential Function Charts (SFC), State charts, Petri Nets, and ordinary object oriented programming languages.

The primary building blocks of Grafchart are steps (representing states) and transitions (representing the change of states). A step and a transition are connected by an arc. An active step is indicated by the presence of a token in the step. Grafchart also supports alternative and parallel branches. An example of a Grafchart application is depicted in Figure 6. Associated with the steps are actions that are executed at certain occasions. To each transition a boolean condition is associated. The transition is enabled when all preceding steps are active. An enabled transition fires if its condition is true, this means that the preceding steps are deactivated and the succeeding steps are activated. Grafchart supports four hierarchical abstractions mechanisms: macro steps, procedure steps, and process steps.

4.2. Execution engine versus visualization

The latest version of Grafchart, i.e. JGrafchart 2.1.1, is basically an Integrated Development Environment (IDE) with interpreted execution. To be able to execute an application it must first be compiled. The compiler checks that the application is valid and it also attaches some additional data

to the application to prepare it for execution. A successfully compiled application can then be executed directly from the IDE in an interpreted manner using the same Java instances as the editor, as well as the additional compilation data. Thus the editor is also re-used as a visualizer.
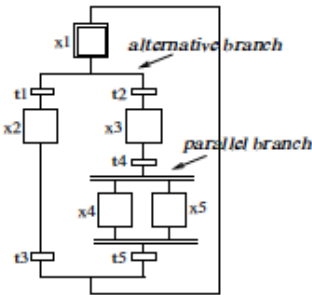

Figure 6: An example of a Grafchart

Implementing visualization in this case is really simple as the editor, the visualizer, and the interpreter are automatically connected. There are however several drawbacks which in an industrial context are likely to be considered showstoppers:

- The execution has to be performed on hardware supporting graphics since the applications can only be executed from the IDE.

- It is not possible to dynamically modify an executing application at runtime since it is not possible to edit and execute at the same time.

- The visualization is tied to the computer that is executing the application.

- Visualization can only be made on one computer.

A different approach would be to let the compiler prepare a separate representation containing all data required for execution. With this approach none of the drawbacks of the current approach are inherent, and the cost is that explicit connections between the editor, the interpreter, and the visualizer(s) are needed.

4.3 SOA-AT with level 2

When connecting to level 2-1, Grafchart has been used with success. Also at this level it is desirable to have one integrated approach that stretches from the process-planning phase, through the commissioning and implementation phase and all the way to the phase for execution of the process control logics. One promising approach is to use the concepts of service-oriented architectures within automation, referred to as SOA-AT. It has been shown how Grafchart can support SOA-AT using DPWS as service technology, and how Grafchart can be used for process modeling and execution in the planning and execution phase at this level (Theorin et al, 2012).

5. FIXED SEMANTICS AND FLEXIBLE SYNTAX

The semantic description of a process provides the basis of an interoperable exchange. The meaning of notations can be described in a machine-understandable manner. Second, the relationships between the notations and the different process modeling languages can be described explicitly. Several technologies to describe services semantically exist (Bhiri, 2008). The most commonly used are SAWSDL and OWL-S.

By using OWL-S as interchange format a decoupling of the syntax and the semantic for process modeling languages can be achieved, see Figure 7. An approach for BPMN was already proofed (Loskyll, M et.al, 2011) and now need to be extended on different process modeling languages.
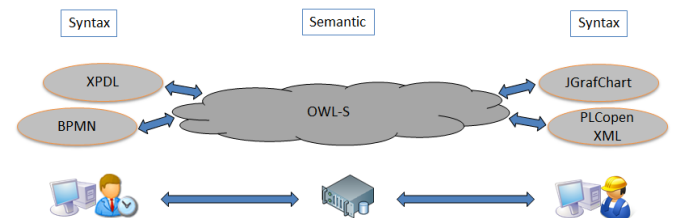

Figure 7: Transfer of the process semantics into OWL-S model.

6. FUTURE RESEARCH

6.1. Time-to-market

In the working procedure frequently used today, different languages are used on the Business level and the Production level, and the objective of the operation phase is different. The design phase is first done on the Business level (level 4). Using a Business language, e.g. BPMN, the process description is illustrated. The process description used in the operations phase has as primary objective to model and visualize the process. Since the language used at the production level (level 3-2) is not the same as the one used at the Business level (level 4), the design phase is redone using another language. The process description used in the operations phase at the production level (level 3-2) has as primary objective to control and execute the actual production. This working procedure is illustrated in Figure 9 (left).
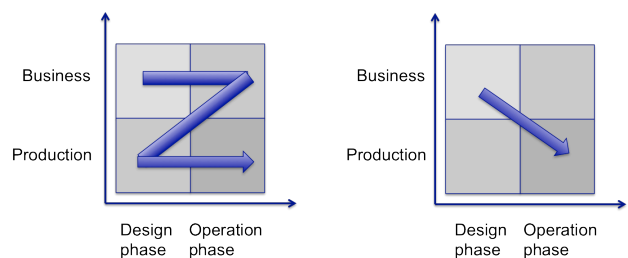

Figure 9: The working procedure of today (left) and a proposed new working procedure (right).

By splitting the execution engine and the visualization engine of Grafchart, the working procedure could be short-cut and the Time-to-market could be shortened. What is done at the Business level could directly also be used at the production level as illustrated in Figure 9 (right).

6.2. Vertical feedback loops

As the vertical integration between the levels becomes more and more important it would be challenging to see, if and what, a unified common graphical language for Business, MES and PCS level, could bring.

A first step would be to investigate if companies of today believe that a vertical integration would be beneficial. The business processes seldom has a connection to the production plant. Various KPIs and aggregated data from the plant are, rarely presented by the mean of a graphical language.

One scenario could be to connect the business processes to the MES. This would imply that the business processes could get data directly from the plant, and thereby directly visualize various KPI for the management, see Figure 10.
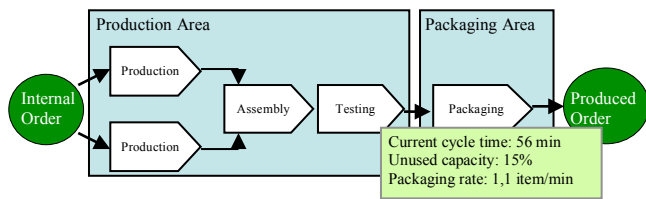

Figure 10: A future scenario regarding inclusion of KPI visualization in a Business process.

6.3. Level of Automation

By splitting the visualization engine and compiler engine of Grafchart it could potentially be possible to allow several visualization tools, e.g., both a visualization that is appreciated when designing business processes and a visualization that is appreciated when executing the production procedures. This would potentially allow the Business processes to, not only be used for modelling and visualization but also for execution, thus increasing the level of automation. To split the visualization engine and compiler engine of Grafchart is therefore an important goal that will also serve to validate the research ideas.

6. CONCLUSIONS

To fulfill increasing requirements in the manufacturing sector, companies are faced with several challenges. Three major challenges have been identified regarding Time-to-market, vertical feedback loops and level of Automation. Grafchart, a graphical language aimed for supervisory control applications, can be used from the process-planning phase, through the implementation phase and all the way to the phase for execution of the process control logics, on the lower levels of the Automation triangle. Work in progress is examining if the same concepts could be used on the higher levels of the Automation triangle as well. By slitting the execution engine and the visualization engine of Grafchart various different visualization tools could potentially be used, however connected by the shared Grafchart semantics. Traditional Business languages, e.g. BPMN, could therefore continue to be used for the process-planning phase whereas traditional production languages can be used for the execution. Since they are connected through the semantics, advantages regarding the three identified challenges could be achieved.

REFERENCES

Aberdeen Group (2006), Forrester, 2000. Global Manufacturing: MES and Beyond Benchmark Report, May 2006.

Arzén, K.-E. (1991), Sequential Function Charts for knowledge-based, real-time applications. In Proc. Third IFAC workshop on AI in Real-time control. Rohnert Park, California, USA

Bhiri, S et al. (2008) *Semantic Web Services for Satisfying SOA Requirements. Advances in Web Semantics I*, spp. 374-395, Springer.

CIMOSA. http://www.cimosa.de (as of 2012-03)

DIN 19226-1 (2010), Control technology - Part 1: General terms and definitions.

Draft-ISO22400-Part1 (2011), Manufacturing operations management —Key performance indicators— Part 1: Overview, concepts and terminology, 2011.

Draft-ISO 22400-Part2 (2011), Manufacturing operations manage-ment —Key performance indicators— Part 2: Definitions and Descriptions of KPIs, 2011.

Draht, R. (2010), *Datenaustausch in der Anlagenplanung mit AutomationML*. Heidelberg, Springer Verlag.

Forrester Research. http://www.forrester.com/(as of 2011-09)

Gorbach G., Nick R. (2002), Collaborative Manufacturing Management Strategies, ARCAdvisory Group, November 2002. Ed. E. Bassett & A. Chatha.

IEC (2003), IEC 62264-1: Enterprise-Control System Integration, Part 1: Models and Terminology. Technical report. International Electrotechical Commission

INCOSE (2007), Model Based Systems Engineering Initivate(MBSE), INCOSE, San Diego, USA, June 2007.

Johnsson, C. (2008), Graphical Languages for Business Processes and Manufacturing Operations, IFAC 2008, Seul, South Korea, July 2008.

Kiefer, J. (2007), *Mechatronikorientierte Planung automatisierter Fertigungszellen im Bereich Karosserierohbau.* Schriftenreihe Produktionstechnik Band 43, Saarland University, Saarbruecken, Germany.

MESA (2006), Metrics that Matter: Uncovering KPIs that Justify Operational Improvements, MESA Interantional October 2006.

OMG (2010), Object Management Group

Theorin, A., Ollinger, L., Johnsson, C. (2012): Service-oriented Process Control with Grafchart and the Device Profile for Web Services, submitted to the 14th IFAC Symposium on Information Control Problems in Manufacturing, INCOM 2012, Bucharest, Romania, may 2012.

Loskyll, M. et al. (2011). Semantic Service *Discovery and Orchestration for Manufacturing Processes.* 16th IEEE International Conference on Emerging Technologies and Factory Automation – 2011 (ETFA'2011) Toulouse, France, 2011.

Pohlmann, E.G. (2008). *Methodik zur prozessorientierten Planung serviceorientierter Fabriksteuerungssysteme.* Fortschritt-Berichte pak 17, Universität Kaiserslautern, Kaiserslautern, Germany.