Proceedings of the 17th World Congress
The International Federation of Automatic Control
Seoul, Korea, July 6-11, 2008

Graphical Languages for Business Processes and Manufacturing Operations

Charlotta Johnsson

Dept of Automatic Control, LTH, Lund University, Sweden
(e-mail: charlotta.johnsson@control.lth.se)

Abstract: The aim of this paper is to present trends, similarities and differences in the usage of graphical languages at the level of Process Control, Manufacturing Operations and Business Systems. The paper also gives ideas of how a common language could be used to increase the integration between the three levels and what advantages this could bring to its user.

1. INTRODUCTION

Many and various decisions have to be taken within an enterprise of today. The decisions can be divided in several layers ranging from long-time strategic decisions to real-time decisions. This is true for all types of enterprises. The decisions concern various types of issues ranging from long-time financial issues, planning, and procurement, through tactical decisions down to operative and real time decisions associated to automated control and human actions, see figure 1. This article focuses upon companies in the manufacturing sector and to issues related to production.

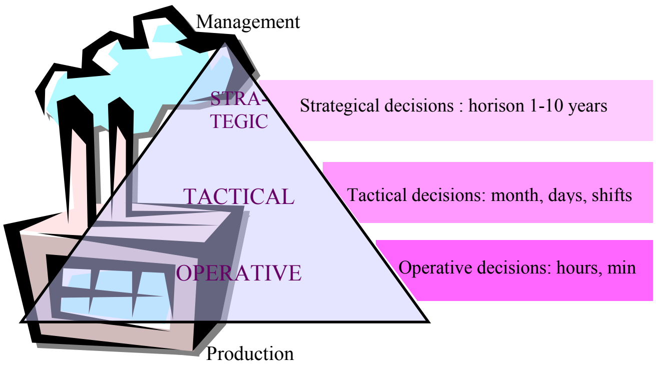

Figure 1: Various decisions within an enterprise and their respective time horizon.

In order to help with the production related decisions, various systems are available. Generally speaking, industrial production systems can be divided into three layers. Closest to the actual process/plant, the process control systems (PCS) reside, with a manufacturing operations system (MOS) often on top of this, and a business system (BS) constituting the superior system, see figure 2.

At the BS level, the system needs to be able to perform reasoning. The system is e.g., responsible for modifying the basic plant production schedule for orders received, based on resource availability changes, energy sources available, power demand levels, and maintenance requirements, etc.

At the MOS level, the system should e.g., calculate the detailed production schedule, perform data collection, calculate various production performance measurements, and assure tracking and traceability.

The PCS level assures classical process control functions such as sensor readings and setting of actuator set-points and PID control of various variables.

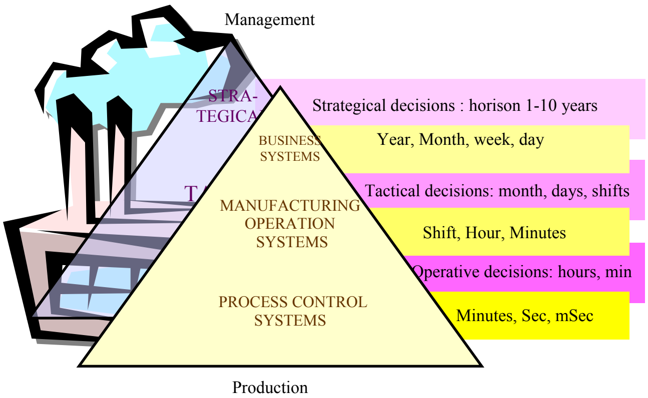

Figure 2: Various decisions and their respective industrial production systems.

As global competition in manufacturing has increased, trends within industry have been to increase the level and amount of control and automation at all three levels. The desire of increasing the integration between the three levels is also notable. Different aspects of control, monitoring, supervision and data and information management have to be dealt with within and between the three levels. This can be assured via different means such as implementation through textual languages, or implementation through graphical languages etc.

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 believe it is easier to program by graphical means then by textual means, increased possibility to use colours, etc. In this spirit, the possibility to have a common, or similar graphical language at all three levels seems very appealing. It would have advantages such as; easier collaboration between the levels, better cross code understanding, use of same terminology etc.

978-1-1234-7890-2/08/$20.00 © 2008 IFAC

10.3182/20080706-5-KR-1001.3026

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 it syntax, semantics and pragmatics.

The aim if this paper is to present trends, similarities and differences in the usage of graphical languages at the BS, MOS and PCS levels. The paper also gives ideas of how a common language could be used to increase the integration between the three levels and what advantages this could bring to its user.

2. HIERARCHIES

The international organisation ISA (Instrument, Systems and Automation Society) has released a standard, named ISA 95 (ISA, 2000), in which, among other things, a physical hierarchy describing how the physical entities within an enterprise are organised. In the physical model, the highest level is the Enterprise, an enterprise consists of one or several Sites, a site consists of one or several Areas. The terminology used for the lower levels varies depending upon the type of industry they apply to (batch, continuous, or discrete). Below the Area we have Process Cells and Units (batch), Production Units and Units (continuous) and Production Line and Work Cells (discrete). This hierarchy is shown in figure 3.

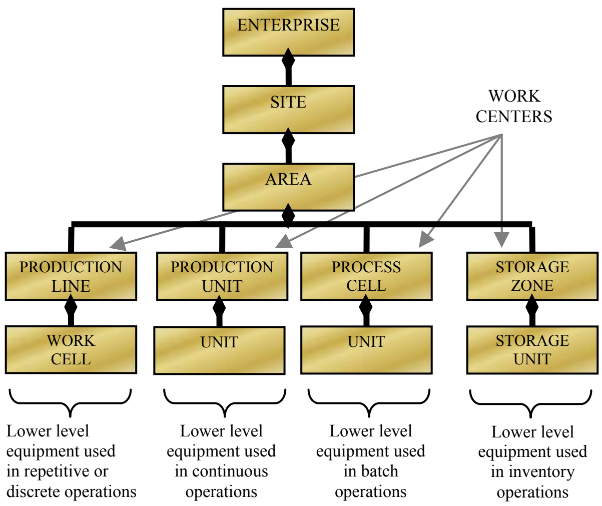

Figure 3: Equipment hierarchy according to the standard ISA 95 (IEC 62264).

The ISA 95 standard also presents a functional hierarchy for the different types of functions within an enterprise. The functional model is used as the basis for the industrial production systems (BS, MOS and PCS) presented in the previous section. The functional hierarchy is based on Purdue Enterprise Reference Model, PERA, defined at Purdue University (Williams, 1992).

3. CURRENT USAGE OF GRAPHICAL LANGUAGES

Roughly speaking one could say that the PCS system should effectuate control of the equipment at the level of and the level below the unit and work cell level. The MOS system should assure the control of the activities residing at the work center, area and/ or site level, and the BS system should have control of the activities at the site and enterprise level. This is highlighted in figure 4.

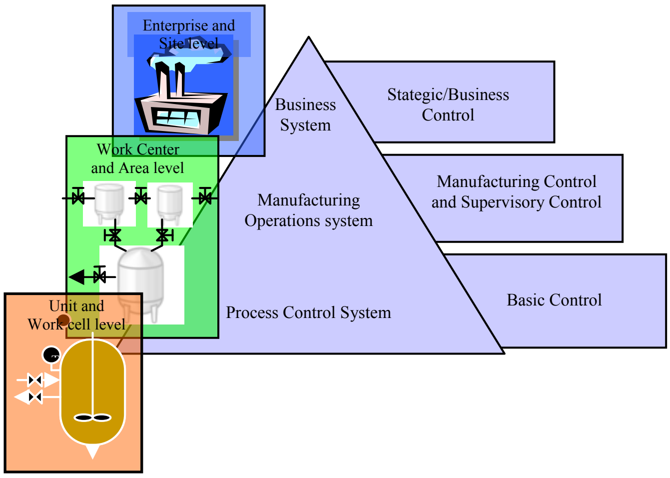

Figure 4: The three control levels and their respective approximate equipment levels.

3.1. Graphical languages at PCS level

The process control system should assure that basic control issues are taken care of, i.e., control issues that are dedicated to establishing and maintaining a specific state of equipment or process condition, and/or simple on/off control. The process control system often depends on the type of industry where it is applied; discrete, continuous or batch.

- Continuous processes are characterized by the production of material in a continuous flow e.g., bulk chemicals, pulp and paper production. In continuous industries the basic control consists of running a system in steady state.
- Discrete processes are characterized by production of a specific quantity of material, where each element of the material can be uniquely identified, e.g., production of cars. In discrete processes the basic control includes the control of start up, transition and shut down phases.
- Batch processes are characterized by generation of finite quantities of material, called a batch, at each production cycle. Material is produced by subjecting specific quantities of input materials to a specified order of processing actions using one or more pieces of equipment. One example could be the production of cookies, in which several cookies are produced in the same batch. The cookies produced in the same batch can not be uniquely identified, but the cookies produced by different batches can be identified. Basic control for batch processes includes characteristics from both continuous and discrete processes.

IEC 61131-3 is a global standard for industrial control programming at the PCS level (IEC, 1993). The standard specifies the syntax and semantics of a five programming languages; Instruction List, Structured Text, Ladder Diagram, Function Block Diagram and Sequential Function Chart (SFC). Two of the languages are textual; Instruction List and Structured Text, and two are graphical; Ladder Diagram and Function Block Diagram. In addition, the graphical language Sequential Function Chart (SFC) is defined with the purpose of structuring the internal organization of a program.

IEC 60848 is another international standard that specifies Grafcet. Grafcet was proposed in France in 1977 as a formal specification and realization method for logical controllers. During several years Grafcet was tested in French industries. It soon proved to be a convenient tool for representing small and medium scale sequential systems at the PCS level. In 1988 Grafcet, with minor changes, was adopted as an international standard known as IEC 60848 (IEC, 1988).

The basic building blocks in Grafcet and SFC are steps and transitions. An example of a Grafcet/SFC is shown in Figure 5.

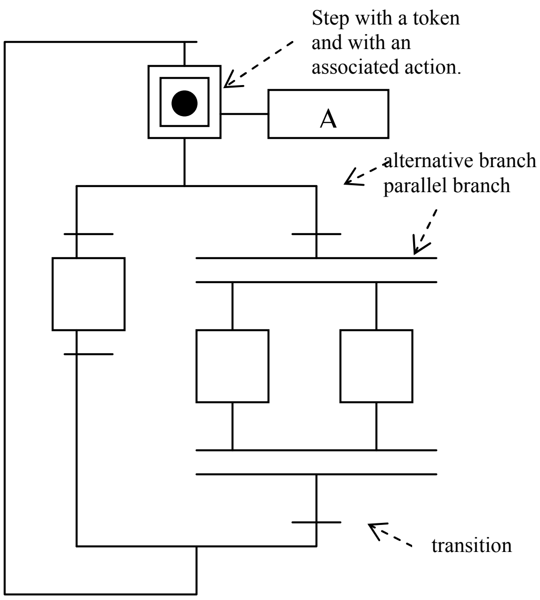

Figure 5: An example of a Grafcet.

In discrete and batch manufacturing, graphical languages such as Grafcet or SFC (Sequential Function Chart) are commonly used. These languages are both based on a state/transition model and can therefore be used in order to express the sequential order in which actions should be carried out. Due to their intuitive nature and their ease of use, they have become very popular. Within continuous industries, textual languages are still most frequently used, however, a sequential language such as Grafcet or SFC can be used for certain tasks.

The traditional user of a PCS system is a person that is highly technically skilled, he/she has a good understanding of the production process and is used to various programming languages.

3.2. Graphical languages at the MOS level

At the MOS level there is a need for both supervisory control at the work cell level and for, what could be called, “manufacturing control” at the area and site level.

Within the batch community a language called Procedural Function Chart (PFC), aimed at supervisory control, has been proposed and included in the ISA standard for batch control called ISA 88.01 (ISA, 1995). This standard has also been accepted as an international standard referred to IEC 61512-1 (IEC, 1997). The PFC takes care of the control of activities at the process cell level. PFC is highly influenced by Grafcet/SFC regarding its syntax. PFC is becoming popular also in the discrete industry.

Another graphical language developed recently and aimed at supervisory control applications at the process cell level is Grafchart (Johnsson, 1999). Grafchart has been developed at Lund Institute of Technology, Sweden since 1991 (Arzén, 1991). Grafchart is based on Grafcet/SFC. It contains support for step with actions, transitions, parallel and alternative branching. In addition to this there is additional support for procedural programming (procedural step) and for separate execution threads (process steps). These two new graphical elements together with the fact that Grafchart is object-oriented and that it has support for parameterization and methods and message passing highly facilitates the reusability of an implementation. Grafchart served as a source of inspiration for PFCs.

An example of a Grafchart, used for structuring a batch recipe at the process cell level, is shown in Figure 6.

The typical user of a MOS system is an operator working in the production plant and/or its production manager. Both persons have a good understanding of the production process and are often technically skilled. In case the production plant is executed by manual actions, the operator might use the graphical language for visualizing the sequential order of the operations he/she should perform. If the production plant is run automatically by various equipments and machines, the operator might use the graphical language for looking at the sequential progress of the execution taking place in the production plant.

The manufacturing operation systems have lately received a lot of interest form the industries. The reason being that there is a need for a clear definition of activities at this level as well as a desire of a more sophisticated and advanced coordination control at this level, let’s call it “manufacturing control”. At this level, activities at the area and/or site level should be taken care of, this could for example be the synchronization of a batch process cell followed by a discrete packaging line.

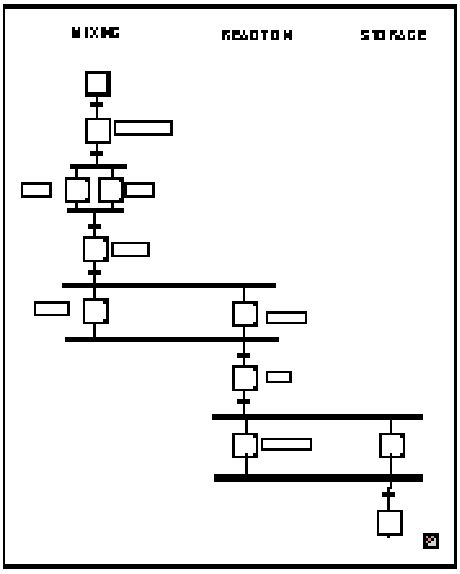

Figure 6: An example of a batch recipe structured with Grafchart.

The ISA 95 part 3 standard specifies eight activities that should occur at the MOS level; resource management, product definition management, detailed production scheduling, production dispatching, production execution management, production tracking, production data collection, and Production Performance Analysis (ISA, 2005). The activities defined within the standard ISA 95 should be considered as a guideline for what activities that may be required within a manufacturing operations system. The standard gives no guidelines about how the activities should be organized, i.e., how the relative order between the activities should be specified or executed. A logical thought could be to have a graphical language by which the sequential order that exists between the activities can be specified. In a proposed new part of the ISA 95 standard the interoperability between the activities described in ISA 95 part 3 will be specified. At the moment, only a data exchange interface is proposed, i.e., no graphical language.

An example of such a language is High-Level Grafchart (Johnsson 1998), (Johnsson 1999). High-Level Grafchart is an extended version of Grafchart. It incorporates ideas from Petri Nets, High-Level Petri nets, and object-oriented programming. In Grafchart as well as in Grafcet/SFC and in PFC, the token is simply a Boolean indicator indicating if the step is active or not. Only one token is allowed to reside in the chart. In the high-level version of Grafchart the tokens are objects with an identity. The tokens are allowed to have attributes and methods. This gives interesting and structuring possibilities. The size of the charts can be reduced without loosing the intuitive understanding of what actions are carried out in the chart.

Another possibility of a language usable within the manufacturing operation and control system could be an extension of the PFCs. However, at the moment, no initiative has been taken by ISA to develop a standard for such a language. Since a standardized language is missing, certain companies have developed their own languages.

The need for a graphical control language will certainly increase with the new trend of Service Oriented Architectures. A service-oriented architecture is essentially a collection of services. These services communicate with each other. The communication can involve either simple data passing or it could involve two or more services coordinating some activity. Some means of connecting services to each other is needed (SOA, 2007). This mean could, with advantage, be a graphical language.

The typical user of a graphical language aimed at "manufacturing control" at the MOS level would be the production manager. The production manager typically has a good understanding of the production process and is technically skilled. However, the production manager also has the need of looking at the production process from a non-technical perspective in order to answer questions like; are we in-schedule? Can we start another production run? What is the current rate-of-occupancy in the plant?

3.3. Graphical languages at the BS level

At the Business Systems level, the interest in depicting business processes has increased. Business Process Managment has been identified as a top business priority (Recker et al., 2006).

Within a producing enterprise, the ultimate level depicts the overall process of going from a customer order to delivery, or from customer need to customer satisfaction. The overall business process consists of one or several sub-process, one of them is certainly manufacturing.

The overall process description is used for off-line visualization of the business process for the management and board of the company.

The manufacturing process, being a sub-part of the overall business process, can also be visualized. Figure 7 shows an example of a manufacturing process, i.e., how to go from internal order to produced order. The manufacturing process could span over one or many areas, e.g., the production area and the packaging area.

In many cases, the language used at this level is most often local to the company and the processes visualized are only used as off-line tool for organization improvements.

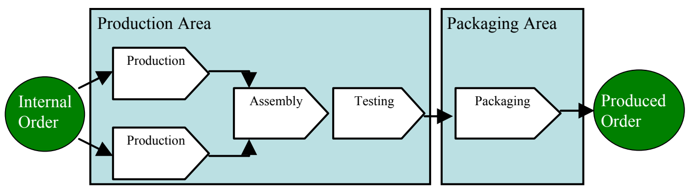

Figure 7: The manufacturing business process

In general, process models at the BS level, serve two main purposes.

First, intuitive business process models are used for scoping the project, and capturing and discussing business requirements and process improvement initiatives with subject matter experts. The recent introduction of legislative frameworks such as the Sarbanes-Oxley Act (Nielsen et al., 2004) further contributed to the increasing interest in business process modeling as a way of capturing and graphically documenting the processes of an organization (Recker et al., 2006).

Second, business process models are used for process automation, which requires their conversion into executable specifications, i.e., turning the depicted processes into an on-line workflow management system (Recker et al., 2006).

The graphical languages for processes are being harmonized, one example of such an initiative is The Business Process Modeling Notation (BPMN) developed by Business Process Management Initiative (BPMI), (BPMI, 2007).

The typical person interested in the business processes are management, i.e., usually persons without a deep technical knowledge of the production process. In addition, the management does more often have a strategic or economic interest in looking at the processes.

3.4. Situation at present

The situation concerning the usage of graphical languages could be summaries in the following way, see also figure 8:

- PCS level: Standardized graphical languages are used for basic control issues.
- MOS level: 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 "Manufacturing control" issues.
- At the PCS and MOS level, the languages are use 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 and MOS level are very similar.
- BS level: Local graphical languages are used for visualising the business processes. Initiatives exist striving to standardize the language used for visualization of business processes.
- At the BS level, 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 the BS level is different from the one used at the PCS and MOS level.

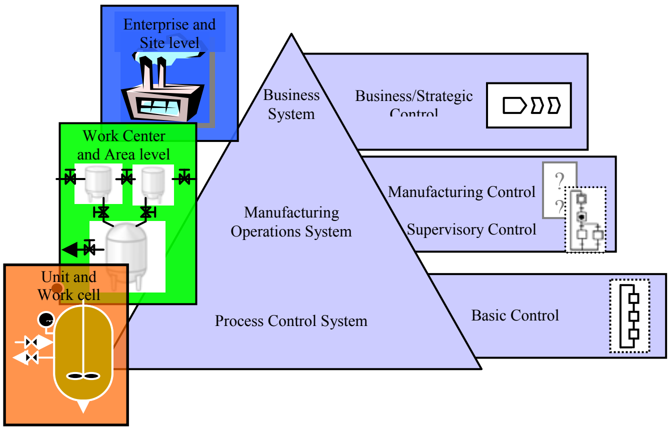

Figure 8: The current situation concerning the usage of graphical languages

4. INTEGRATION

As global competition in manufacturing has increased, trends within industry have been to increase the level and amount of control and automation at all three levels. The requirement to increasing the integration between the three levels is also notable, being driven by business needs and business processes, see figure 9.

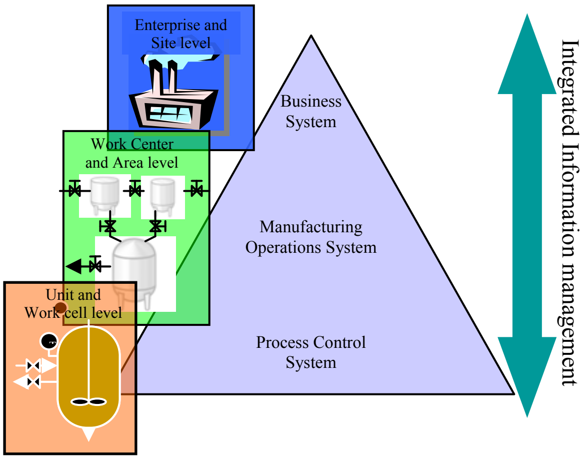

Figure 9: The need for vertical integrated information management has increased lately.

There are always some business processes that needs information from production or needs to exercise control over production. Typical business drivers of today are; Just in time production (requires detailed knowledge about the available capacity in the plant), asset efficiency and overall equipment efficiency (requires detailed knowledge of actual use), reduced cycle time (requires detailed knowledge about start and stop times), new regulations for tracabilty (requires better logging and tracking possibilities), finding hidden potentials (requires the collection and analysis of previously unknown data), etc.

Another approach to integration is Unified Enterprise Modeling Language, UEML. However, UEML is not intended as a programming language by itself but rather as an intermediate language, or a hub, through which different languages can be connected (UEML, 2007).

4.1. Future research

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 BS, MOS and PCS level, could bring.

A first step would be to investigate if companies of today believe that a vertical integration would be beneficial. One scenario could be to connect the business processes to the Manufacturing Operations Systems. This would imply that the business processes could get data directly from the plant, and thereby directly visualize various Key Performance Indicators for the management, see figure 10.

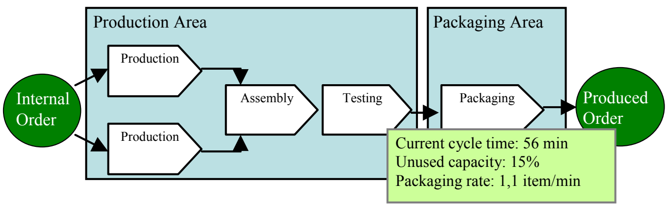

Figure 10: A future scenario

6. CONCLUSIONS

The focus for this article is graphical languages used at companies in the producing and manufacturing sector and to issues related to production. The production systems used within the companiy can be divided in three hierarchical levels; Process Control Systems, Manufacturing Operations Systems and Business Systems. The trends, similarities and differences in the usage of graphical languages at these are examined.

There are strong similarities between the languages used at the PCS level for basic control and the languages used at the MOS level for supervisory control. At these levels, the activities at the unit level and below are taken care of, and there is a very close connection to the production occurring in the plant.

To my knowledge, there is currently no standardized graphical language for "manufacturing control", i.e., for coordinating and synchronizing the activities between various work centers and area.

At the BS level, the business processes are modelled and executed. Initiatives exist that strive to standardize the graphical language used at this level. The syntax is different from the one used at the PCS and MOS level. 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.

It would be very interesting to further investigate, if, to what extent and how, the graphical languages used at the three levels could be harmonized, and what advantages this could bring to its user.

REFERENCES

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.

BPMI (2007), http://www.bpmi.org/

IEC (1988), IEC 60848: Preparation of function charts for control systems. Technical report. International Electrotechical Commission.

IEC (1993), IEC 61131: Programmable controllers – part 3: Programming languages. Technical report. International Electrotechical Commission.

IEC (1997), IEC 61512-1: Batch Control, Part 1: Models and Terminology. Technical report. International Electrotechical Commission.

ISA (1995), ISA-S88.01, Batch Control, Part 1: Models and Terminology. Instrumentation, Systems and Automation Society.

ISA (2000), ISA-95.00.01, Enterprise-Control System Integration Part 1: Models and Terminology. Instrumentation, Systems and Automation Society.

ISA (2005), ISA-95.00.03, Enterprise-Control System Integration Part 3: Models of Manufacturing Operations Management. Instrumentation, Systems and Automation Society.

Johnsson, C. (1998), On batch recipe structures using high-level Grafchart. In Symposium ADPM'98 – Automation of mixed processes: Dynamical and hybrid systems. Reims, France.

Johnsson, C. (1999), A Graphical Language for Batch Control. PhD-thesis, Lund Institute of Technology, Lund, Sweden.

Nielsen and Main, (2004), Company Liability After the Sarbanes-Oxley Act, Insights (18) 10, pp. 2-12.

Recker J., Rosemann M., Indulska M. and Green P. (2006), http://is.tm.tue.nl/staff/wvdaalst/BPMcenter/reports/2006/BPM-06-20.pdf BPM Center Report BPM-06-20, BPMcenter.org.

SOA (2007), http://www.service-architecture.com

UEML (2007), http://is.uib.no/wiki/UEML/UEML

Williams, T. (1992), The Purdue Enterprise Reference Architecture--A Technical Guide for CIM Planning and Implementation, ISA, Research Triangle Park, NC, USA.