Automatic Generation of Object Models for Process Planning and Control Purposes using an International standard for Information Exchange

Petter Falkman[*], **Johan Nielsen**[†], **Bengt Lennartson**[*]

[*] Control and Automation Laboratory, Department of Signals and Systems
Chalmers University of Technology, Göteborg, Sweden
pf, bl@s2.chalmers.se

[†] Computer Systems for Design and Manufacturing, Department of Production Engineering
Kungliga Tekniska Högskolan, Stockholm, Sweden
jni@iip.kth.se

Abstract

In this paper a formal mapping between static information models and dynamic models is presented. The static information models are given according to an international standard for product, process and resource information exchange, (ISO 10303-214). The dynamic models are described as Discrete Event Systems. The product, process and resource information is automatically converted into product routes and used for simulation, controller synthesis and verification. A high level language, combining Petri nets and process algebra, is presented and used for specification of desired routes. A main implication of the presented method is that it enables the reuse of process information when creating dynamic models for process control. This method also enables simulation and verification to be conducted early in the development chain.

Keywords– Modelling methods, information modelling, Petri Nets, Process algebra.

1 Introduction

In order to be competitive, engineering companies of today have to be flexible and responsive to rapidly changing market needs. For this reason, it is important for companies to decrease the time to market while still maintaining or increasing product quality, all at a low cost. A step towards decreasing the time to market is a more efficient information exchange between product and manufacturing systems design.

Making the information exchange more efficient means that information about product design solutions becomes instantly available to engineers involved in the manufacturing system design. More concretely a process planner will start the documentation of how to manufacture a product based on preliminary design solutions already during the product design. If information can be made instantly available to engineers, the iteration cycle between product and manufacturing systems design can also be made shorter.

This process documentation will be used as a base for simulating how the introduction of a new product will affect an existing, or new, manufacturing system. The outcome of the simulation will influence the final design solution of the product as well as the manufacturing system. The created dynamic models will, in addition to this, also be used for verification and automatic controller synthesis.

This paper focuses on verification and automatic controller synthesis. The controller synthesis includes two levels of control descriptions. First, the resource allocation system and second, a more detailed control of specific applications, e.g. control of a robot cell. In the resource allocation system a number of products utilize a number of shared resources which are to be booked and unbooked. A high level language intended to simplify the specification of desired routes is presented here. This modelling language combines Petri nets and process algebra in order to achieve compact representations of the product routes. The more detailed control descriptions involves specific control

for each resource and can be seen as a decomposition from the higher level resource allocation system which is not dealt with in this paper. The focus in this paper is, however, on the resource allocation system.

The process plan defines process information as a set of product, process and resource characteristics, defining what to produce, and how it should be done. This information can be created using several different systems, such as CAD-systems, Robot simulation and Off-Line Programming (OLP) systems.

In order to automatically generate dynamic process models for process control purposes, a mapping is necessary. This mapping should define the relationship between the static information and the dynamic process models.

Much research has already been conducted on information and discrete event modelling, e.g. [10], [8], and [3] discussing information modelling and [2], [6], and [7] discussing discrete event modeling. However, little has been investigated concerning the connection between information and dynamic models, i.e. how an information structure could be mapped to the dynamic structure of a process plan. The purpose of this paper therefor is threefold: first to capture the requirements of the information structure, second to capture the requirements of the dynamic structure, and third to show how the mapping between the information and dynamic structure could be realized.

The information structure is given according to the ISO10303-214 or the STEP-standard (STandard for Exchange of Product model data). The mapping has been achieved by analyzing the information structure, (ISO 10303-214), and the dynamic structure, (the MPPN-model) which was introduced in [4], to gain knowledge of the semantics of their respective objects and structures. The gained knowledge has then been synthesized to result in the semi-formally defined mapping model. Finally, the result has been validated using a case study at Scania Oskarshamn, Sweden. This has been done by populating the ISO 10303-214 model with data from the Scania case, and then implementing the mapping method in order to automatically create an MPPN-model based on the Scania data. The case is presented below and developed throughout the paper.

Example 1 – A robot cell in a Scania factory *The robot cell shown in Figure 1 consists of six resources, a robot R, a gripper G, a welding machine W, two output buffers B_1 and B_2, an input buffer I, an operator O, and two fixtures left FL and right FR.*

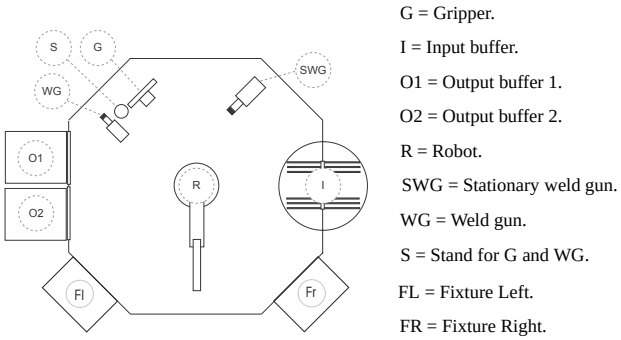

Figure 1: A manufacturing cell.

The Scania robot cell involves two main processes, **StationaryWelding** *and* **RobotWelding**.

StationaryWelding *As input to the robot cell there are geometrically welded plates placed on the turn table. StationaryWelding involves three sub-processes:* **Get**, *robot and gripper is used to get workpiece from instation.* **Weld**, *robot, gripper, and stationary weld gun is used to weld approximately 30 weld spots.* **Put**, *robot and gripper is used to put the workpiece in outstation $O1$ or $O2$.*

RobotWelding *As input to the robot cell an operator places the geometrically welded plates in one of two fixtures. RobotWelding also involves three sub-processes: The first is* **Place** *and involves an operator placing a workpiece on either the right or the left fixture. The second is* **Weld** *where the robot, the weld gun, and one of the fixtures LF or RF is used to weld about 30 weld spots and finally the third and final process* **Put** *where the robot together with the gripper is used for putting the workpiece in outstation $O1$ or $O2$.*

□

In the following two sections an introduction to both the MPPN modelling language and the ISO 10303-214 standard is presented. In addition to this a comparison between the static and dynamic models is made with respect to the product, process, and manufacturing resource (PPR) representation in both MPPN and ISO 10303-214 (AP214).

2 Mixed Process algebra Petri Net

The MPPN language combines Petri nets and process algebra in order to create product specifications. The MPPN language uses process operators for alternative, synchronization in order to realize compact specifications.

Process operators

The transition between two Petri net places in an MPPN is a process P. A process $P = a_1 \rightarrow P_1$ describes that, first the event a_1 occurs, then it behaves like a process P_1.

Alternative The alternative operator $+$ specifies that there is a choice between two processes. Let two processes be defined as $P = a_1 \rightarrow P_1$ and $Q = b_1 \rightarrow Q_1$. Then an alternative between these two processes is described as

$$P + Q = a_1 \rightarrow P_1 \mid b_1 \rightarrow Q_1 \tag{1}$$

using Hoare's [6] choice symbol $|$. This implies that either event a_1 occurs followed by process P_1 or event b_1 occurs followed by process Q_1.

Synchronization The nonstandard synchronization operator $\&$ implies that one or more processes are to be synchronized, with no respect to common events, and executed in parallel. Similar ideas for event synchronization can be found in [1]. Again, consider two processes $P = a_1 \rightarrow P_1$ and $Q = b_1 \rightarrow Q_1$. The synchronization operator & can be described as

$$P\&Q = a_1\&b_1 \rightarrow P_1\&Q_1 \tag{2}$$

This means that a_1 in P occurs at the same time as b_1 in Q. This synchronized event is denoted $a_1\&b_1$. The synchronization operator $\&$ is useful when *flexibility* and *reusability* is desired.

Parallel processes Parallel processes are defined using the Petri net constructs instead of introducing a parallel process operator. This is done in order to preserve a good graphical presentation of the modelled system. In Figure 2 there are two processes P and Q which are to be executed in parallel.

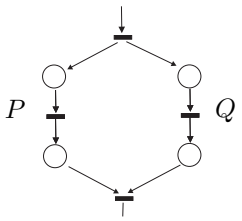

Figure 2: Two processes P and Q are executed in parallel.

Product Model

The dynamic model of a product is the process model that will be described in the next section. However, the static information for a product, e.g. product id, may, in the MPPN model, be assigned to a token. The number of tokens control the number of products manufactured and the number of products or product parts that are allowed in to the manufacturing system at the same time. Note that this may involve colored Petri nets but this extension is not emphasized in this paper.

Process Model

The *resource allocation system* involves a set of products that share a set of resources within a manufacturing system. To ensure that only one product at a time is using a specific resource it is necessary for each resource to be *booked* by a specific product. It is also important to control that there are no *blocking* or *deadlock states*. A routing specification specifies a products route through a resource system and may be described on two levels:

- a *high level routing specification* (HRS) that describes which processes an object are to undergo, in which order these processes are to be executed, and which resource(s) that may be used for each individual process.

- a *booking and unbooking specification*, which describes on a more detailed level how the shared resources are to be booked and unbooked, based on the HRS, to obtain the desired route through the resource system.

Process Operation (PO) In the resource allocation system the transition between two places in the HRS is a *process operation* (PO). A PO involves two processes, a booking process B and an unbooking process U. A booking and unbooking model is automatically created given an HRS.

Example 2 – Robot cell *An HRS to the left in Figure 3 specifies the three processes involved for the Stationary-Welding, in the robot cell example. The first PO requires three resources: the robot R, the gripper G and the Instation I. The second resource requires the robot R, the gripper G and the weld gun WG. To the right in Figure 3 is described on a more detailed level how the resources involved are to be booked and unbooked. Note that resources that are required in several operations are not unbooked.*

□

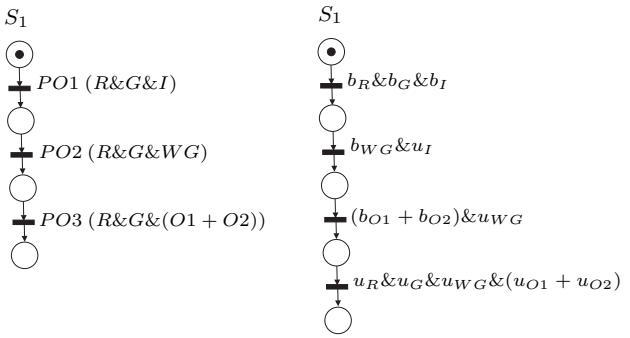

Figure 3: A routing specification is given as an HRS to the left and a booking and unbooking specification to the right. Note that resources that are required in more than one operation in a row are not unbooked.

Manufacturing System Model

A model of the manufacturing system, for the resource booking system, is created by synchronizing all of the involved resource models.

3 The STEP AP214 Model

In this section the product, process, and manufacturing resource (PPR) representation in ISO 10303-214 (AP214) will be described. The objective is to describe where to find the information needed to generate the MPPN-model. Before the PPR-model is described a brief introduction to ISO 10303 (STEP) and the EXPRESS language will be given.

ISO 10303 STEP is an international standard that "provides a representation of product information along with the necessary mechanisms and definitions to enable product data to be exchanged" [11]. The term exchange should here be interpreted as the exchange of data between computer systems in environments associated with the complete life-cycle of a product, including manufacturing.

The standard consists of different parts, called application protocols, that define the scope, context, and information requirements for a particular application, e.g. the automotive industry (AP214), or electrical design and installation (AP212).

EXPRESS language The EXPRESS language is a formally specified and structured language [9] used to define the application interpreted models in STEP. The EXPRESS is an earlier and more general alternative to UML. The basic constructs of EXPRESS or EXPRESS-G (a graphical subset of EXPRESS) is the entity and the attribute. An entity is similar to an object in object-oriented programming.

Graphically, in EXPRESS-G, an entity is represented as a box with a name written in it. Attributes are represented by a line ending with a small circle, showing the direction of the relationship. They are labelled with the name of the attribute as well as any cardinality constraints. A dashed line represents an optional attribute, whereas a thick line represents a supertype-subtype relationship, i.e. the same as inheritance in object-oriented programming.

Product Model

The most important product entities in AP214 is the *item* (i) and *item_version* (iv). These are the holders of product meta-data, such as identifiers, version data, classification data and much more.

Process Model

The process model in AP214, cf. Figure 4, has a central role in the generation of the MPPN-model. It is the holder of all the necessary process information, such as plan identifier, relationships between processes etc.

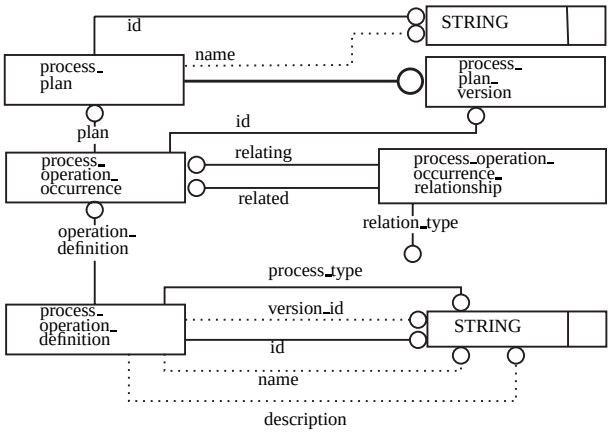

Figure 4: Representation of process data in AP214.

The process model consists of a structure to hold meta-data about a process plan. This structure is identified by the *process_plan* (pp) and *process_plan_version* (ppv) in Figure 4. A *process_plan*, consists of one or more processes represented by the *process_operation_occurrence* (poo). The *process_operation_occurrence* represents the occurrence of a process in a process plan. More specifically, it represents the occurrence of a definition of a process, the *process_operation_definition* (pod). This mechanism enables the reuse of a definition in several different places in a process plan as well as in several differ-

ent process plans and thus, different versions of a plan can reuse definitions that have not been changed from a former version. For instance, alternative resource for the same operation would be represented by the same definition but with different resources assigned to different *process_operation_occurrence*s all representing the same definition.

The structure of the process plan is represented on the *process_operation_occurrence* level, i.e. the level where sequence, alternative, simultaneity, and substitution relationships between processes are represented. This relationship is represented by the *process_operation_occurrence_relationship* (poor) where the attribute *relation_type* holds the type of relationship. The attribute *relating* points in the direction of the *process_operation_occurrence* prior to the *process_operation_occurrence* pointed out by the attribute *related*.

Manufacturing System Model

Manufacturing resources can be represented in several different ways in AP214, depending on the level of detail and the design life cycle stage.
The *single_instance* and *physical_instance* are both instances of an abstract representation of a manufacturing resource (*item*), but there is one significant difference. The *single_instance* represents an occurrence of a type of manufacturing resource whereas the *physical_instance* represents a physical resource on the shop floor. Thus the *single_instance* is better used for planning purposes before a physical resource exists and the *physical_instance* is better used when there already exists a physical resource.

4 Mapping of AP214 into MPPN

In this section a description of the relationship between AP214 and the MPPN product, process and manufacturing system models is given. Examples are given in order to illustrate the mapping of the static description of the product to be manufactured into a dynamic model using the graphical notation of both descriptions. The actual algorithm is not shown mainly due to lack of space. A simplified algorithm is given in [5].

Product

For the purpose of creating a product in the MPPN-model only a product identifier is needed. The product identifier is represented by the *item_version.id* in AP214. This identifier will be assigned to the *token* in MPPN which implies a colored extension of the MPPN. The product identifier is related to process information via the *process_plan.produced_output*.

Manufacturing system

The attributes of the manufacturing resources that are needed in order to generate a MPPN-model are the *single_instance.id* and *physical_instance.serial_number*.

Process

Process information needed in order to create an MPPN-model is *process_plan.id*, *process_operation_occurrence.id*, *process_operation_definition.id*, and information about relationships between processes. Table 1 describes the use of AP214 process model information when creating an MPPN-model.

AP214 Information	MPPN-model
Plan (pp) identifier	Routing specification identifier
Occurrence (poo) identifier	Petri net place (pnp) identifier
Definition (pod) identifier	Process operator (PO) identifier
Relationship (poor) type	Net structure

Table 1: The use of AP214 information when creating a MPPN-model.

The differences between *product_operation_definition* and *product_operation_occurrence* are several but the most important one in this paper is that the *product_operation_definition* gives a general description of what an operation involves, whereas the *product_operation_occurence* describes on a more detailed level which resource to use in a specific operation. For instance, a *product_operation_definition* may be executed by two different *product_operation_occurences* in that that they use different resources. In the MPPN model a process that only differs in which resource they require is regarded as the same *process operation*. This means that it is natural to use the *product_operation_definition* identification when translating into MPPNs.

Alternative resources

In an earlier phase of this project alternative resources has already been implemented. In the STEP standard alternative resource is represented by *process_operation_occurences* that refers to the same *process_operation_definition* and also refers to each other with the *process_operation_occurence_relationship* attribute *relation_type* equal to 'alternative', cf Figure 4. In the MPPN

the alternative resource is represented by a number of alternative resources in the transition equation, cf 5.

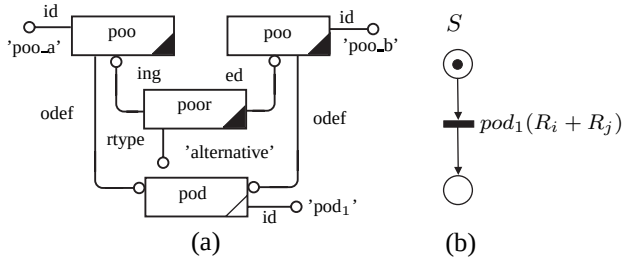

Figure 5: Alternative process given as a High Level Routing specification (b) and an instantiated STEP AP214-model (a). This is alternative processes in the STEP standard with *process_operation_occurences* refereing to the same *process_operation_definition*. Each *process_operation_occurence* relates to one resource each R_1 and R_2

In figure 5(a) a small part of a populated STEP model is represented. There are two *process_operation_occurences*, with respective id 'poo_a' and 'poo_b', which refers to each other with the *relation_type* attribute 'alternative'. Both of the *process_operation_occurences* also refer to the same *process_operation_definition* pod_1. This implies that the same operation is being executed by both *process_operation_occurences*, however these two *process_operation_occurences* are referring to different resources which is not shown in Figure 5(a). In Figure 5(b) an HRS is described showing the use of the + operator for representation of the resource choice.

Alternative processes

Alternative, or split, differs from the earlier mentioned alternative resource in that it in STEP are *process_operation_occurences* that refer to each other with the *process_operation_occurence_relationship* with the attribute *relation_type* equal to 'alternative', cf Figure 4, but the different *process_operation_occurences* do not refer to the same *process_operation_definition*. This results in a split of the sequence in the MPPN in to two or more separate branches, cf Figure 6.

Parallel

Parallel processes are described in the MPPN in the same manner as in ordinary Petri Nets, c.f. Figure 2. In step this is modelled as two or more *process_operation_occurences* that refer to each other with the *process_operation_occurence_relationship* with the attribute *relation_type* equal to 'parallel', cf Figure 4.

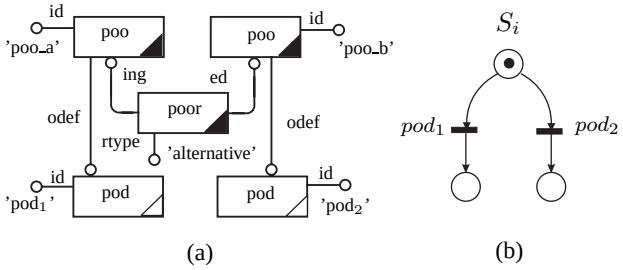

Figure 6: Alternative process given as a High Level Routing specification (b) and an instantiated STEP AP214-model (a). Note that this is different from alternative resources which in STEP is described by the fact that the involved processes are refereing to the same definition.

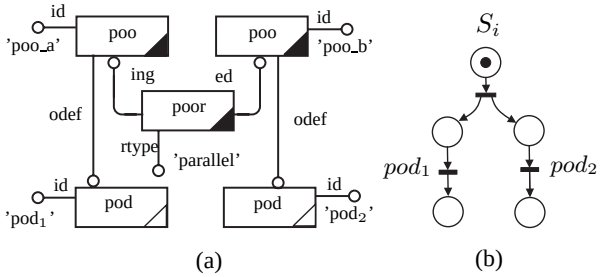

Figure 7: Parallel process given as a High Level Routing specification (b) and an instantiated STEP AP214-model (a).

Example 3 – Robot cell *In Figure 8 shows how the* StationaryWleding *process in the Scania robot cell may be modelled with STEP. The same process is in Figure 9 described using the High level part of the MPPN language. As can be seen in Figure 8 there are four* process_operation_occurences *and only three* process_operation_definitions. *Two* process_operation_occurences *are referring to each other with the relation type 'alternative'. Transferring the STEP model in Figure 8 into the MPPN model in Figure 9results in an MPPN with three operations executed in sequence. The three operations required a set of resources each and as a consequence of the two* process_operation_occurences *referring to the same* process_operation_definitions *exists alternative resources for* $pod3$.

□

The presented ideas so far involve only part of all the information needed in order to control a single cell or a whole plant. There is security information and much more but at this point we have concentrated on translating flow control information.

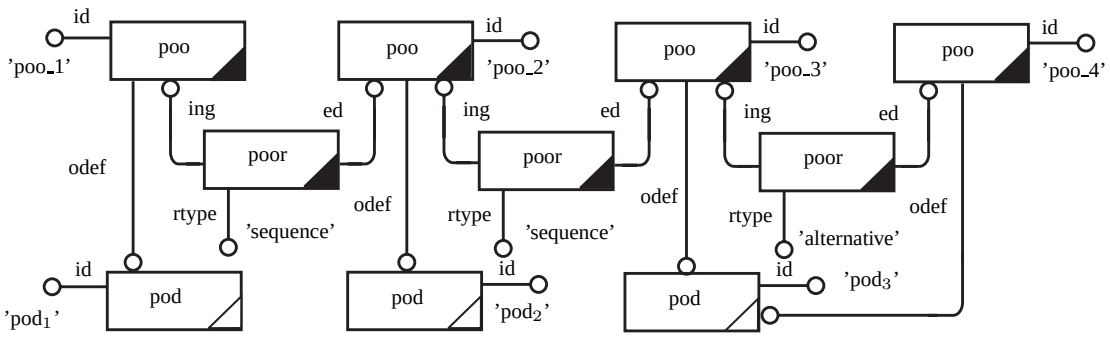

Figure 8: Instantiated example of the *StationaryWelding* process of the Scania robot cell given as a simplified instantiated STEP AP214 model. Note that the each *process_operation_occurence* relates to one or more resources but for clarity this is not shown in this picture.

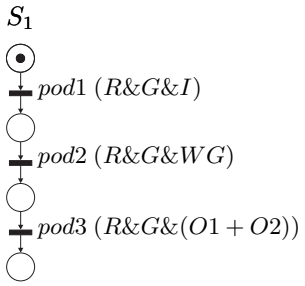

Figure 9: A routing specification given as an HRS and describing the *StationaryWelding* process of the Scania Robot cell.

5 Conclusions and Future work

The suggested method implies a reliable framework for the exchange of control related information, which involves resource, product and process information. In addition to this it delivers the expected information fast, which is crucial when short lead times are required.

One of the main advantages of this method is that it involves, and uses, a well accepted international standard, STEP AP214, for the exchange of product, process, and resource related information.

In future work the entire method will be implemented and applied on a large industry case.

References

[1] A. Arnold. *Finite Transition Systems: Semantics of Communicating Systems*. International Series in Computer Science. Prentice–Hall International, Englewood Cliffs, NJ, 1994.

[2] C.G. Cassandras and S. Lafortune. *Introduction to Discrete Event Systems*. Kluwer Academic Publishers, 1999.

[3] W. Eversheim, G. Marczinski, and R. Cremer. Structured modelling of manufacturing processes as nc-data preparation. In Annals of the CIRP, volume 40/1, pages 429-432, 1991.

[4] P. Falkman and B. Lennartson. Combined process algebra and petri nets for specification of resource booking problems. In *Proc. of 2001 IEEE American Control Conference*, Arlington, VA, USA, June 2001.

[5] P. Falkman, J. Nielsen, and B. Lennartson. A formal mapping of static information models into dynamic models for process planning and control purposes. In *Proc. of WODES 2002*, Spain, Oct 2002.

[6] C.A.R. Hoare. *Communicating Sequential Processes*. International Series in Computer Science. Prentice–Hall International, Englewood Cliffs, NJ, 1985.

[7] P.J. Ramadge and W.M. Wonham. Supervisory control of a class of discrete event processes. *SIAM J. Control Optim.*, 25(1):206–230, Jan 1987.

[8] A. Scheller. Information modeling for distributed applications. In *Proc of second IEEE Workshop on Future Trends of Distributed Computing Systems*, 1990.

[9] D.A. Schenck and P.R. Wilson. *Information Modeling: The EXPRESS Way*. Oxford University Press, 1994. ISBN: 0-19-508714-3.

[10] D.A. Schenk and P.R. Wilson. *Information Modeling: The EXPRESS Way*. ISBN 0-19-508714-3. Oxford University Press, 1994.

[11] ISO TC184/SC4. Iso 10303-1: Industrial automation systems and integration - product data representation - and exchange - part 1: Overview and fundamental principles. ISO Standard, 1994.