# Object-oriented engineering data exchange as a base for automatic generation of simulation models

Mike Barth[1], Martin Strube[1], Alexander Fay[1], Peter Weber[2], Jürgen Greifeneder[2]
[1]Helmut-Schmidt-University Hamburg, Germany / [2]ABB Corporate Research Center Germany
{mike.barth, martin.strube, alexander.fay}@hsu-hh.de / {peter.weber, juergen.greifeneder}@de.abb.com

***Abstract* – This document describes an approach to support an automated generation of simulation models by use of already existing engineering data. The concept enables engineers to execute tests in parallel to the development of the control applications, using a physical model of the controlled system. The simulation model forms a virtual representation of the process plant that is still under construction at that time. Both simulation approaches, Hardware-in-the-loop (HIL) or System Simulation, can be applied with varying intensity during the test phases. The piping and instrumentation diagram (P&ID) as an essential design document within engineering processes serves as the data source for model generation. Nowadays, P&IDs are still exchanged between the different project partners on a paper base or portable document format (PDF) respectively. Thus a computer-aided further processing is not possible. In this context the authors use and extend an XML-based data exchange format to export all information from the P&ID that is required for the generation and parameterization of a physical simulation model. With this approach, the object-oriented structure available in some of today's computer aided engineering (CAE) tools can be transferred into a generic representation and be re-used for automatic model generation.**

## I. Introduction

Increasing product complexity and competitive demands on production flexibility challenge automation suppliers to continuously improve their engineering efficiency while keeping a high level of product and system quality.

In this context, plant simulation can offer advantages especially in assuring the quality demands on the resulting artifacts. However, building a simulation model is as individual as building the real plant. To avoid costly activities due to the use of simulation, an automated model generation is desirable. In order to achieve this, a prerequisite is the integration of the model generation process into the engineering tool chain. Especially a tighter connection to data sources like P&I diagrams as well as to various information sources of the detailed engineering process is necessary. The problem of missing standard formats for data exchange will also be addressed in this paper. Such a standard would support the transfer of the required information in a tool-independent way.

Besides the introduction of new techniques for higher production engineering efficiency also the engineering process itself has to be taken into consideration. Inspiration can be taken from several sources, as for example adopting object oriented approaches [1]. Modern product development tends to minimize iterations within engineering stages by integration of early test phases. Thereby the minimization of poor quality costs has priority. The earlier an engineering mistake can be detected the less cost intensive its removal will be - see Fig. 1. [2]

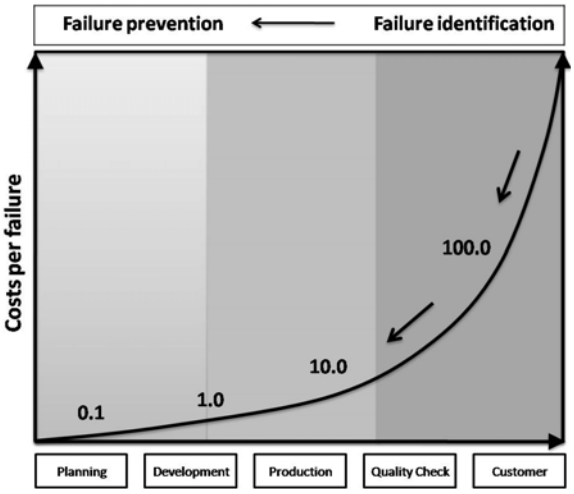


Fig. 1. “Rule of ten” concerning prevention and identification of failures within the product development [2]

Though this correlation between failure cost and failure indication is well known, it is very little reflected in today's engineering procedures. Up to now testing of control code for process control systems (PCS) is mostly put into the in-house “Factory Acceptance Test” (FAT) as well as into the on-site “Site Acceptance Test” (SAT), both rather late compared to the start-up of the plant.

### 1. *State-of-the-art within software based testing of control code*

Dependent on to the respective PCS application, different tools and methods for control code testing are available to the engineers. In this article the focus is on software-supported test concepts. Nevertheless, it has to be pointed out that up to now also hardware (HW) based testing equipment, as for example switchboards and potentiometers are applied to set input signals to the control system. At the same time, computer-aided “Signal Forcing” forms the first out of four SW-supported test variants. Signal Forcing is used especially for single object testing, e.g. testing of control valves. Another variant implements simulation fragments directly in the respective control function block (typical) or integrated into a separate simulation block. The implementation of the simulation logic is done similarly to PLC programming using a language of the IEC (International Electrotechnical Commission) 61131-3 [3], [4] and covers basically discrete models. In addition, commercial simulation environments as for example Simulink are used. These tools represent the physical process by use of signal flow-oriented block diagrams, which are manually developed and modeled.

On the scale of cost intensive proprietary simulation approaches and high fidelity simulation products there is a potential for low cost variants, e.g. based on MS Excel. This is mainly because MS Excel is anyhow very often used by engineers for various tasks mainly in the area of bulk data management. In addition there is no extra license fee to be paid for MS Excel as it is already an integral part of the engineering tool landscape. This tool is also increasingly applied for test activities. A current example for this is discussed in [5]: The use of Excel as a simulation engine for object-based control tests. It relates to a simulation environment particularly developed for semi-automatic generation of simulation models based on the evaluation of PLC engineering data. This approach forms one of the first steps towards an automated generation of physical simulation models.

### 2. *Simulation mode and model variants*

Independently of the selected test tool, the deployment of simulation determines a classification concerning the methods of simulation as well as the methods of modeling. To guarantee a temporal and physical independence of the test objects from the real process plant components within early engineering phases, the approach of a system simulation [6] will be realized first. In this case, both the process control system and the controlled plant are virtual components. The test of the control code operates on the basis of "Soft-controllers" (PC-based emulators) which are executable on a standard PC just like the operator panels. By the end of the application development, in which different disciplines develop their respective applications in parallel processes, the results are integrated into an overall system. At this point the system simulation shifts to a "hardware-in-the-loop" approach (HIL) [7], in which the "Soft-controllers" are substituted by the real process control system. The substitution of the virtual by the physical controls can take place sequentially, one component after the other.

Apart from the type of simulation technique being applied, the tests scenarios can further be differentiated based on the selected kind of modeling complexity. Easiest is to model, simulate and test each plant element isolated from all the other elements. A pump for example receives a start signal from the controller and delivers a feedback signal according to the internal behavior as defined in its model. Today, this is the level of complexity most of the test cases are applied on. At a higher level of modeling the mutual influences between elements are covered. Thereby isolated functional units as well as combinations of the same and even the whole plant may be object of the test scenarios. This is necessary to test sequences which act on different actuators and receive information from several sensors. As an example, engineers might want to test if a pump is still running while a valve in its headwater is closed, which could cause dry running in practice. To test relations like the one above, it is necessary to use a model which contains information, as well as material and other energy flows. With a manual construction of the simulation model, this variant is complex to realize, especially considering the tight scheduling of the engineers and their everyday's work load. For this reason the focus has to be set on an automated generation of the simulation model.

## II. MOTIVATION FOR AUTOMATED MODELING

Concurrent engineering describes an approach of organizing tasks – originally performed sequentially – partly in parallel [8]. Following this idea, an automated model generation offers a potential for optimization because test activities could be done in parallel to the implementation of the control code. As mentioned in chapter I, control code testing is an essential part of control engineering but there is no possibility to carry out these tests in combination with real plant within early project phases. Therefore simulation is often used as a tool for evaluating the developed control code. Due to the virtual character of a system simulation and its inherent independence of any hardware equipment, simulation based tests can already be launched during application development.

### 1. *Arguments for automated model generation*

In the following some of the most important arguments for automatic model generation are listed:

- The simulation model has to be continuously adapted to the current state of the project as well as to the current P&ID version and therefore be recreated several times.
- Manual modeling tasks could result in an incorrect virtual plant model and therefore cause misleading test results. It is necessary that the quality of the simulation model doesn't need to be tested itself.
- Additional engineering effort for the development of a simulation model is not acceptable for PCS system engineers.
- Domain-specialists for simulation, modeling and control engineering would be necessary for the development of suitable test models.

In order to emphasize this argumentation the following study, describing the importance of model building in relation to the different simulation phases, is considered.

### 2. *Importance of modeling for simulation studies*

The VDI guideline 3633 Part 1 – simulation of systems in materials handling, logistics and production – structures a simulation study in different steps and phases. Fig. 2 shows the workflow of a simulation study with the iteration loop caused by changes during the plant development process [9].

A special problem of the early control code testing is that the development of the controlled system has not been finished yet. That means changes of the physical properties are most likely and there will be more than one iteration step for the simulation. As Fig. 2 shows there are two time consuming steps necessary for preparation before a simulation session can begin: the data collection and the model generation.

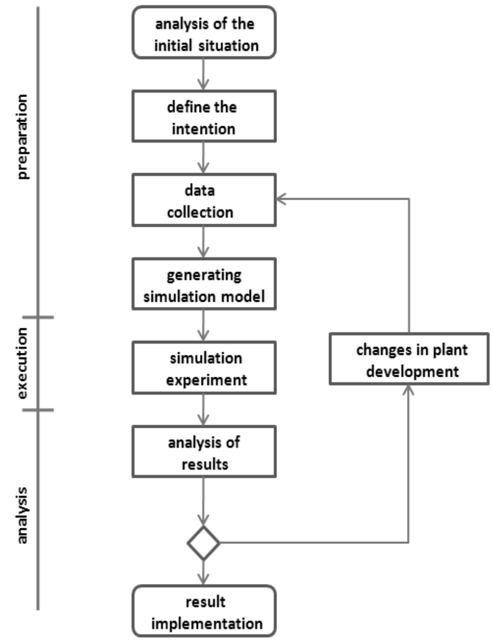


Fig. 2. Workflow of a simulation study

### 3. *Problems of manual modeling and data collection*

The planning process of plants and other engineering systems is characterized by interactions and information exchanges between different engineering disciplines. Nowadays nearly every engineering domain uses specific tools and document types during the planning process. That is one of the reasons why simulation experts need a lot of time to collect the required information for modeling. This proceeding is not only time consuming but also error prone because it has to be ensured that the collected information is up-to-date and consistent.

Although there are many simulation tools on the market which use model templates or library concepts to decrease the coding effort, it is still a time consuming process to type in the necessary model parameters manually [10].

The aim must be a reduction of the model building effort and an increase of the model quality. Therefore it is required to identify the data sources out of which all relevant information can be taken. To use this for an automated generation of the simulation model, it needs to be stored in a suitable exchange format.

## III. DATA SOURCE FOR MODELING INFORMATION

Nowadays the use of CAE tools for planning and design of process plants is a common practice. These CAE tools can be distinguished according to their historical development into object-oriented, database-oriented and document-oriented tools. When the planning process went from the drawing board to the digital world, the first CAE tools came onto the market. These document-oriented systems focused on the creation of drawings. In parallel the database-oriented systems had been developed for collecting alphanumeric information about the properties and functionalities of equipments and machines. Object-oriented CAE tools represent the newest generation that tries to unify the advantages of both other methods while using objects to merge the graphic character together with the alphanumeric attributes. The following analysis demonstrates the capabilities of object-oriented CAE tools in regard to the automated generation of physical simulation models for process plants.

One central document in the planning of process plants is the piping and instrumentation diagram (P&ID) that represents a symbolic illustration of the plant devices, the equipment of measurement and control technology and the connection between them. The P&ID acts as a "living" document which is used during the whole lifecycle of a plant. Most object-oriented CAE tools enable a simplified creation of the P&ID by the use of templates in form of object-libraries or object-databases for most relevant components. To create the P&ID, it is necessary to instantiate parameterize and connect these templates which manage the graphical symbol and the specific attributes for the equipment in form of objects. The connection between the graphical symbol of equipment and its attributes makes the P&ID to become a central data collector for engineers in different disciplines.

Fig. 3 shows how this philosophy can be realized within a CAE tool illustrated by a cut-off of the P&ID that shows the graphical representation of a tank and its properties.

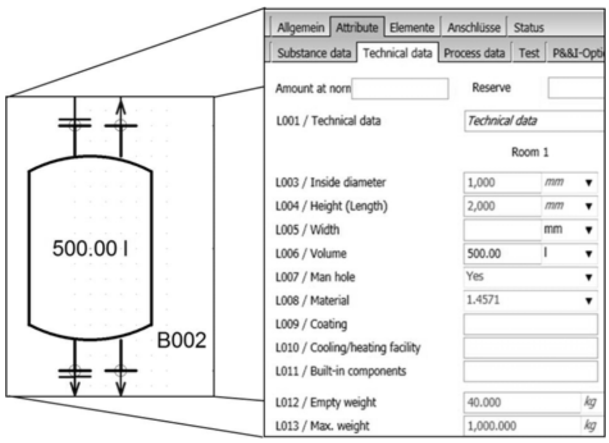


Fig. 3. P&I representation of a batch vessel and corresponding parameterization face-plate

The possibility to find all information that is needed for generating a physical plant model in the P&ID helps to reduce the effort for data collection described in chapter II. Following this it is not necessary to collect the required information out of engineering-tools and documents from the co-operating departments. Another advantage of object-oriented CAE tools is the bidirectional data exchange between different engineering departments that helps to increase data consistency and actuality. Being able to extract information out of the P&ID does not only support the initial generation of a simulation model but also ensures that changes done to these data can immediately be updated in the simulation model. Most plant engineering tools offer functionalities to export configuration data. Therefore it is an obvious approach to export the information from the P&ID's, which is required to build up a physical model of the plant.

Preconditions for the automated generation of plant simulation models are (1) to automate the data collection process, (2) to find a way to export this information into a tool-independent data exchange format and (3) to define a suitable information structure that follows the idea of object-oriented data processing.

## IV. DATA EXCHANGE STRUCTURE FOR THE GENERATION OF SIMULATION MODELS

The object-oriented approach of modern CAE tools described in chapter III forms one of the bases for the automated generation of simulation models. One of the goals is to make this structure usable for succeeding procedures of model generation by means of a manufacturer and system-independent data exchange format. The problem of suitable data exchange formats exists since computer-aided engineering became widespread. The introduction of standards was mainly driven by the car industry that used 2D- later also 3D-CAD-systems for construction. Already 1979 Boeing, General Electric and NIST developed the IGES format (Initial Graphic Exchange Standard) [11], which was limited on geometry information however. 1984 followed by STEP (Standard For The Exchange Of Product Data) [12] that did not essentially differ from IGES except for an extension for finite elements analysis. 1990 one of the existing STEP-variants was further developed to be the first data exchange format for process plants and therefore resulted in the ISO 10303 part 227 “Plant spatial configuration” [13]. Other stages of development followed. The increasing importance and need for a continuous documentation of change management activities formed the developing impulse for ISO 15926 "Life cycle data for process plant" [14], [15], [16]. This format concentrates primarily on the data exchange between CAE tools, as well as the archiving of component-based plant changes during production.

For the automated generation of a simulation model a data exchange format is needed, which structures the P&I-data independently of CAE tools and stores it in a flexible and expandable way. 2005 the International Electronical Commission published the meta-data exchange format "CAEX" (Computer Aided Engineering eXchange) [17] as part of PAS 62424. CAEX fulfills the formulated requirements, while it forms a XML-based class structure built upon three library concepts. Using this format, a complete information transfer of P&I-object data and their connections can be achieved. The plant hierarchy, which can also be transferred via CAEX, maintains both parameterizations made in the CAE tool and its material, information and energy flow connections. In combination with a likewise object-oriented simulation language that bases on a library concept – including physically modeled plant objects – virtual sensors and actuators can be instantiated, parameterized and connected.

### *1. Advantages of stand-alone P&I data export*

The direct consumption of P&ID information has substantial advantages compared to the use of general CAE tool project data:

- The information does not differ from what is to be seen on a printing of the P&ID.
- No critical project data, like manufacturer lists, addresses, budgets, prescriptions, calculations, specifications, etc. are exported from the CAE tool.
- In the case of an additional change within the P&ID only the data of the affected plant segment has to be exported.
- The data set of the CAEX file becomes smaller and contains merely that information which is needed for a later simulation model generation.
- The versioning of the CAEX file can be derived from the versioning of the P&ID.

According to this, the export functionality of P&IDs into a suitable data exchange format can be compared to their physical print-out or the conversion into a PDF document (Portable Document Format). Usually the PCS system supplier gets the PDF documents from its customer.

### *2. Modeling information in CAEX*

It is not intended to explain the layout of CAEX in detail. For this reason it is referred to [17], [18], [19]. However, it shall be mentioned, that an integral component of CAEX is that all the parts of the model (plant objects, its attributes and interfaces) are flexibly expandable – not only the model as a whole. Furthermore, features like object orientation as well as library concepts are implemented. The use of a library is already well known by engineers that operate simulation environments, like SIMULINK [20], Dymola [21], etc. The classification in

- Interface library,
- Role library and
- System Unit library

makes it possible to map physical and equation based modeled objects, as it is done for example in the simulation language Modelica [22].

The batch vessel illustrated in Fig. 4 is used as an explanatory example. The system “vessel” is defined on the functional level based on the chosen role class which includes appropriate attributes and interface information. Default values can be already given to those attributes. This is especially needed for parameters that cannot be taken over immediately from the P&ID, but are necessary for a complete definition of the models in a physical manner.

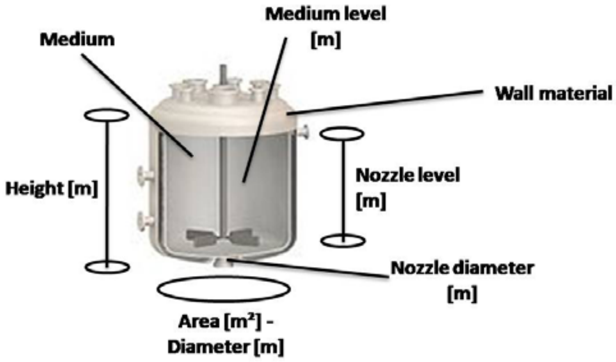


Fig. 4. Batch vessel with simulation parameters

The interfaces included in the P&ID regarding the material, energy and information flow are defined in an "Interface Library". These can be mapped directly on simulation objects by the use of physically described models.

Table 1 lists an example for each of the mentioned interfaces again on the vessel system.

TABLE I
COMPARISON OF INTERFACES OF SIMULATION AND P&ID

| | Interface type | | |
|---|---|---|---|
| | Material flow | Energy flow | Information flow |
| P&ID | Nozzle | Heat Pipe | Sensor |
| Simulation model | Liquid flow connector | Heat transfer connector | Data Port (int, real, bool) |

In consensus with the modeling of the simulation library [23], the third library class "System Unit Class" makes it possible to define more complex plant components which can contain plant elements or units itself. This approach is comparable with the product and module structures reflected in PLC typicals created by automation suppliers. Thus units can be summarized, e.g. a vessel with assembled filling state and pressure sensors, as well as flanged mixing engines.

The final configuration of the arrangement to be simulated is carried out in the "Instance Hierarchy". Here the classes stored in the different libraries are instantiated, parameterized and connected among each other on the basis of the P&ID information. As shown in Fig. 5 the batch vessel from Fig. 4 will be instantiated as B002 and parameterized based on the information of the shown CAE face-plate. For the purpose of a better overview, the details of the tank model are shown in a "XML Grid View". The XML notation has not been changed for that reason.

InternalElement (12)

| | Name | RefBas... | ID | PhysicalAttribute |
|---|---|---|---|---|
| 1 | B002 | Components/Library/Ressources/MaterialStorageClosed/Components | {f49d48fc-f14a-4011-ba5a-088c3952d3da} | PhysicalAttribute (10) |

| | Name | Attr... | D... | Value |
|---|---|---|---|---|
| 1 | Height | double | | 2,0 |
| 2 | Inside_Diameter | double | | 1,0 |
| 3 | Level_Max | double | | 1,8 |
| 4 | Volume | double | | 1,4 |
| 5 | Wall_Thickness | double | | 0,05 |
| 6 | Man_Hole | bool | | true |
| 7 | Empty_Weight | double | | 15,0 |
| 8 | Max_Weight | double | | 100,0 |
| 9 | Medium | string | | 1.4310 |
| 10 | T_Max | double | | 400,0 |

Fig. 5. XML grid view of element attributes

In order to link the elements available as instances in accordance with their material, information and energy flow, so called "Internal Links" are built. Fig. 6 shows an excerpt of a material flow internal link section. Line 1 has to be read as follows: The unique name for this connection is "B001 – Pump001". Names should be as self-explanatory as possible, i.e. this link describes the connection from the batch vessel "B001" to the pump "Pump001". On the left hand side (RefPartnerSideA), i.e. at the vessel, this link is connected to the "Forward-port" (Material Output) of "B001". On the right hand side (RefPartnerSideB) it is connected to the "Backward-port" of (Material Input) of "Pump001".

InternalLink
InternalLink (11)

| | Name | RefPartnerSideA | RefPartnerSideB |
|---|---|---|---|
| 1 | B001 - Pump001 | Forward:MO B001 | Backward:MI Pump001 |
| 2 | Pumpe001 - Junction01 | Forward:MO Pump001 | Backward:MI Junction01 |
| 3 | Junction01 - Y002 | Forward:MO1 Junction01 | Backward:MI Y002 |
| 4 | Junction01 - Y003 | Forward:MO2 Junction01 | Backward:MI Y003 |
| 5 | Y002 - B002 | Forward:MO Y002 | Backward:MI B002 |
| 6 | Y003 - B003 | Forward:MO Y003 | Backward:MI B003 |
| 7 | B002 - F003 | Forward:MO B002 | Backward:MI F003 |
| 8 | B003 - Y004 | Forward:MO B003 | Backward:MI Y004 |
| 9 | F003 - Junction02 | Forward:MO F003 | Backward:MI1 Junction02 |
| 10 | Y004 - Junction02 | Forward:MO Y004 | Backward:MI2 Junction02 |
| 11 | Junction02 - B001 | Forward:MO Junction01 | Backward:MI B001 |

Fig. 6. XML grid view of internal links

It has to be recognized that every object owns at least one material output (MO) and at least one material input (MI). If an object has several material in- or outputs (e.g. Junction02 follows F003 and Y004), it typically represents a material storage element (e.g. batch vessel) or a pipe branch / join element respectively.

### 3. *Subsequent use of the plant description*

The plant data that is available in CAEX can be used for the automated generation of a simulation model in this form. The system and vendor independent XML notation realizes different programming techniques for model generating algorithms in adaption of the respective simulation system or modeling language being used. Similar attempts can be conceived for the initialization of the required OPC based communication between the controller and the simulation environment. In this case PCS engineering data is evaluated to automatically initialize OPC-Tags. In order to detail the design of a simulation model it is also possible to enrich the CAE data by PCS specific object information such as signal ranges or units.

## V. Conclusion

The combination of product quality and time to market decides in its balance decisively on the success of an automation solution. Considering this, the continuous lowering of fault rates by early identification of failures forms a challenge for all system suppliers. Using the data exchange format CAEX as an information source for the generation of physical simulation models is a further step contributing in this direction. The subsequent advantages are shorter quality test cycles and finally an earlier start of production, as well as a more accurate automation solution with lower costs for fault corrections.

## References


[1] D.N. Ramos-Hernandez, P.J. Fleming, S. Bennett, S. Hope, J.M. Bass, M.J. Baxter, “Process Control Systems Integration Using Object Oriented Technology,” pp.148, Technology of Object-Oriented Languages and Systems, 2001.

[2] T. Pfeiffer, „Quality Management: Strategies, Methods, Techniques”, Vol. 3, pp. xxviii ; Carl-Hanser Verlag München, Wien, 2002.

[3] International Electrotechnical Commission: IEC, 61131 Part 3: Programmable controlles - Programming languages, 2003.

[4] International Electrotechnical Commission: IEC, 61131 Part 5: Programmable controlles - Communications, 2001.

[5] M. Barth; P. Weber; A. Fay; J. Greifeneder, “Semiautomatic generated element and process simulation models for PLC tests,“ Proceedings of “Automation 2009”, Baden-Baden, Germany, 2009. (in german language)

[6] D. Abel, “Integration of Advanced Control in the process industry, Rapid Control Prototyping,” Wiley-VCH Verlag Weinheim, 2008 (in german language).

[7] G. Pritschow, S. Röck, “Hardware in the Loop Simulation of Machine Tools,” CIRP Annals, 52(1), pp.259-298, 2004.

[8] A.N. Pidgeon, M.E. Begin, J. Eickhoff, J. de Kruyf, “System Simulation and Verification Facility,” Proceedings of the DASIA 98 - Data Systems in Aerospace, Athens, Greece, 1998.

[9] VDI guideline 3633 Page 1 “simulation of systems in materials handling, logistics and production,” March 2003.

[10] I. Paprotny, W. Zhao, G.T. Mackulak, “Reducing model creation cycle time by automated conversion of a CAD AHMS layout design Simulation,” Conference Proceedings Winter Simulation Conference Phoenix, Arizona, United States, Volume 1, pp. 779 – 783, 1999.

[11] B. Goldstein, S. Kemmerer, C. Parks, "A Brief History of Early Product Data Exchange Standards," NISTIR 6221, September 1998.

[12] ISO 10303-1, “Overview and Fundamental Principles. Industrial Automation Systems and Integration – Product Data Representation and Exchange,” Part 1, 1994.

[13] ISO 10303-227, “Application Protocol: Plant Spatial Configuration. Industrial Automation Systems and Integration - Product Data Representation and Exchange”, Part 227, 2003.

[14] ISO 15926-1, “Overview and Fundamental Principles. Industrial Automation Systems and Integration - Oil and Gas,” Part 1, 2003.

[15] ISO 15926-1, “Data model. Industrial Automation Systems and Integration - Oil and Gas,” Part 2, 2003.

[16] D. Leal, “ISO 15926 Life Cycle Data for Process Plant: An Overview,” Oil & Gas Science and Technology – Rev. IFP, Vol. 60, No. 4, pp. 629-637; Institut français du pétrole; 2005.

[17] International Electrotechnical Commission: IEC, PAS 62424 – “Representation of process control engineering requests in P&I diagrams and data exchange between P&ID tools and PCE-CAE tools”.

[18] S.Y. Yim, H. G. Ananthakumar, L. Benabbas, A. Horch, R. Drath, N.F. Thornhill, “Using the Process Schematic in Plant-wide Disturbance Analysis,” ESCAPE-16 & PSE-2006, Garmisch-Partenkirchen, Germany, July 10-14.2006.

[19] M. Fedai, R. Draht, “CAEX - a neutral data exchange format for engineering data,” International Automation Technology 01/2005, 3, pp.43-51.

[20] W.J. Palm, “Introduction to MATLAB 7 for engineers,” Edition: 2, revised, illustrated; McGraw Hill Professional, 2004.

[21] Dynasim AB: Dymola 7.1 release notes - Lund, September 30, 2008 - www.dynasim.se/Dymola71release.pdf (last check: 17.03.2009).

[22] P. Fritzson, “Principles of object-oriented modeling and simulation with Modelica 2.1,” Institute of Electrical and Electronics Engineers, Inc. Wiley Interscience, 2004.

[23] H. Elmqvist, H. Tummescheit; M. Otter, “Object-Oriented Modeling of Thermo-Fluid Systems,” Proceedings of the 3rd International Modelica Conference, Linköping, November 3-4, 2003.