Ontology Approach for Co-Use of 3D Plant Modelling and Large Scale Process Simulation

Marko Luukkainen, Tommi Karhela
VTT Technical Research Centre of Finland, P.O.Box 1000, FIN-02044 VTT
marko.luukkainen@vtt.fi

Abstract

Traditionally plant design- and process simulation applications are separated, but increasing use of simulation to support design and validation drives to find means for integration. We propose an ontology-based method for combining 3D plant modelling, large scale process simulation, and visualization. The use of ontologies allows clear distinction of a plant model and a simulation model, which enables multiple simulation models and simulation types to be used with the same plant model. Another contribution is that the same user interface and same visualization techniques can be used with different simulators.

Keywords: 3D Industrial plant modelling, process simulation, visualization, ontology

1. Introduction

Use of simulation has increased in process industry. Plants are getting more complicated, and need for advanced methods for complex system design is growing. Today it is common that when new plant is designed, designs are tested and verified using simulators. Typical simulators contain their own user interfaces and their own data structures, and transforming modelling data to simulator compatible form is seldom automatic. Usually a plant model has to be redone for the simulator. This makes the use of simulators more laborious and more error prone. In typical case, various systems in plant are modelled with different software, which further increases the risk of error.

Currently, the most common way to create a model for process simulation is to use two dimensional diagrams. At the same time, nearly all new plants and changes to old plants are designed with some 3D modelling software. Combining 3D modelling with simulation combines advantages from both: Two-dimensional diagrams cannot show physical properties of the modelled process, but 3D models are similar to real world objects. 3D-modelling comes especially handy when sizes of objects are needed by the simulator. In 2D-diagrams numbers must be given explicitly, whereas in 3D-modelling all dimensional values are already in the modelled process.

Because simulators are used for testing and verifying the functionality of the modelled process, it is more practical to include simulation within the plant modelling software. This would allow an engineer to test his design earlier and improve his understanding of the modelled process and its behaviour. The improved understanding leads to better design practices and results in savings in time and money. In understanding the process and its behaviour, visualization is important. Animations and other visualization techniques can improve perception of simulation results over text-based numeric information or separate 2D graphs.

Ontology-based knowledge representation has gained popularity because of its flexibility and expressive power. Here using the ontology-based approach has critical advantages: 3D modelling user interface does not have to be bound to any particular simulator nor any particular simulator type. Instead, simulators can be linked to 3D model's data structures, and the same visualization package can be used with many different simulators and the same 3D model with multiple simulators. This is required, because different simulators are used in different stages of the modelling process: First mass and energy balances are simulated using steady state simulation; later dynamic simulation can be used for various purposes, including testing plant's behaviour in changing conditions, and in operator training.

2. Related work

Gruber[1] uses ontologies for knowledge sharing. Uschold and Gruninger [2] use them for communication, inter-operability, and systems engineering. According to Guarino [3], ontologies can be used as database components, user interface components, and application program components either development time or run-time. Ontologies have become popular in several scientific areas including Artificial Intelligence, Computational Linguistics, and Database Theory.

Kalogerakis, Christodoulakis, and Moumoutzis [4] present an interoperable framework for integration of virtual reality scenes and semantic information. Their system uses several different semantic mappings between information model and graphical model. For instance, “equivalence”-relation describes that every NURBS curve represents a membrane or a polygon mesh represents a component in a specific shell. Linking graphics to domain concepts works with domain independent mapping relations and intermediate mapping classes, which allow interpreting relationships always the same way. Park [5] uses ontologies to combine dynamic models used in simulation to geometric models of the phenomena being modelled, calling the methodology as integrative multimodeling. With integrative multimodelling, the user could bind geometric models to simulation model, and then visualize the simulation. Lehtonen and Karhela [6] combine process simulation and active diagrams with ontologies. The system used ontologies and rules to generate simulation model from graphical model (diagram), and then visualizes the simulation with animations.

Rohrer [7] shows how visualization provides better means to understand behaviour of simulated model. The main reason is that the models of manufacturing simulations tend to be complex and counter intuitive, but visualization is a natural way to communicate a lot of information between computer and its user. Quick et al. [8] develops a system for manufacturing simulations and visualizations. The system contains a library of reusable models of equipment and animation methods. The user could use those to create manually or in some cases automatically a visualization model for the simulation. Animations are bound to event-based simulation and it could visualize for example movements of a forklift.

3. Ontologies and mappings

Before we can create an ontology for 3D plant modelling, we have to choose base ontology; the ontology that describes concepts, like classes and instances, basic concepts that are needed in information modelling. Instead of using existing ontology languages, we have developed our own language called Layer0. Layer0 is similar to OWL Full (Web Ontology Language) [9], but while OWL uses open-world definitions, where ontologies can contribute new concepts to other ontologies, Layer0 uses closed world semantics. This enables more stable environment, where concepts remain the same. Other Layer0 specific features are separation of real-time data from modelling data, which is used with simulation engines.

Our purpose is to distinct graphical model of a plant from its simulation models. At the same time we need a way to describe 3D graphical models of plant equipment and their animations. Instead of solving these separately, it is beneficial to use common representation for 3D graphics. We use X3D [10] specification as a basis for our ontology. Using ontology gives better ways to describe relationships between objects and properties than X3D; therefore its concepts are not just copied to ontological format. Basic design principle in 3D modelling ontology is that it supports only those features that are necessary, and other features can be added by creating new ontologies. This goes according to the ontology design principles [1], where minimal ontological commitment is considered as good feature that enhances re-usability of ontologies. Therefore created base ontology for 3D graphics contains only minimal set of the features that X3D supports.

Created 3D graphics ontology is used as basis for CSG modelling ontology (Constructive Solid Geometry) and 3D plant modelling ontology. CSG modelling is used for describing shape of equipment, and while it restricts freedom for 3D modelling compared to polygon modelling or B-Rep modelling (Boundary representation), it provides all necessary features for equipment modelling.

Similarly as 3D graphics ontology, concepts of 3D plant modelling ontology are based on previous taxonomies. Here we use combination of names and concepts from commercial plant modelling products and from ISO 15926 [11,12], but in very reduced form. For example, ISO 15926 defines thousands of concepts that are related to process industry, but using all of them would complicate our design without giving any real benefits. The plant modelling ontology is extended from the 3D modelling ontology, giving it 3D modelling capabilities.

Since the main purpose 3D plant modelling is to act as a user interface to process simulation, the simulation model must be generated according to the 3D model. Our purpose is not to create our own simulator, but instead, in the long run, use existing simulators, for instance simulators developed at VTT [13]. The principle here is that an ontology, which describes data structures of a simulator, is created for each used simulator, and then, with mappings, system will automatically generate simulation model when the user creates 3D model. As general, mappings can be divided to one-to-one mappings, where single concepts are linked to each other. For example diameter of pipes in 3D model is linked to diameter in simulation model. More difficult cases are one-to-many and many-to-many mappings, which link a group of concepts to each other. One good example is that when 3D model is linked to simulation model, multiple pipelines in 3D could be described as single connection in simulation model. More detailed discussion can be found in [14].

4. Visualizations

The most intuitive way to visualize behaviour of equipment is either mimicking their natural behaviour or to use colours to represent some information about them.

For instance, visualizing tank fill with an animation that changes liquid's height inside the tank is natural to every user and requires no further explanation (Figure 1). Sometimes this is not enough, for example, while status of a valve can be visualized by animating the position of its handle, the handle will most likely be too small to be seen from long distances. Because understanding of the behaviour of the simulated process requires that some part of the process can be seen at the same time, it may be more practical to visualize state of the valve by changing its colour. Since the colour of equipment can visualize multiple different properties, the user must be responsible for binding the animation to a simulation value. Animations bound to natural features are different, for instance, the fluid height changing animation is only useful for visualizing fluid height.

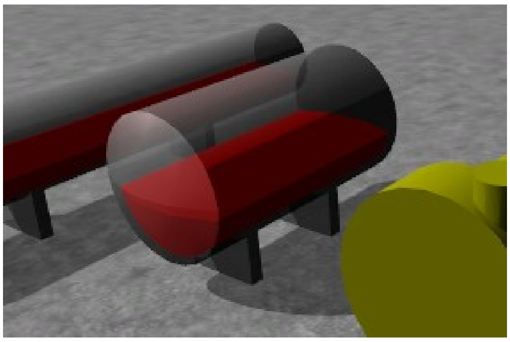

Figure 1. 3D model of a tank visualizes liquid level.

Animating just equipment will not bring complete visualization of a process plant: also flows inside pipes are in critical part. Therefore we need means to visualize flows. Similarly as with equipment, flows may have multiple properties that the user may want to visualize. The most common of them is probably mass flow, but other properties, like temperature and pressure are also important. Various techniques exists for flow visualization in 2D and in 3D [15], but most of them are not usable in visualizing a flow in pipe, which is essentially 1D flow in 3D space. Only techniques that we considered useful were glyphs and particle animations. Glyphs have proven useful in depicting spatially complex, multivariate data, because their graphical properties, such as colour and size, can be bound to visualized data. They can be seen in weather forecasts in TV, where arrows indicate wind direction and temperature. Particle animations are probably more realistic visualization of flow, since flow velocity can directly be mapped to particle velocity, it requires constant updates of positions of particles, which can be computationally demanding.

Figure 2 shows overall data model, including a 3D plant model, references from plant model to equipment models in a library, and a simulation model generated from the plant model. The figure also shows how animations can be chosen for each piece of equipment and how their animation can be bound to simulation values.

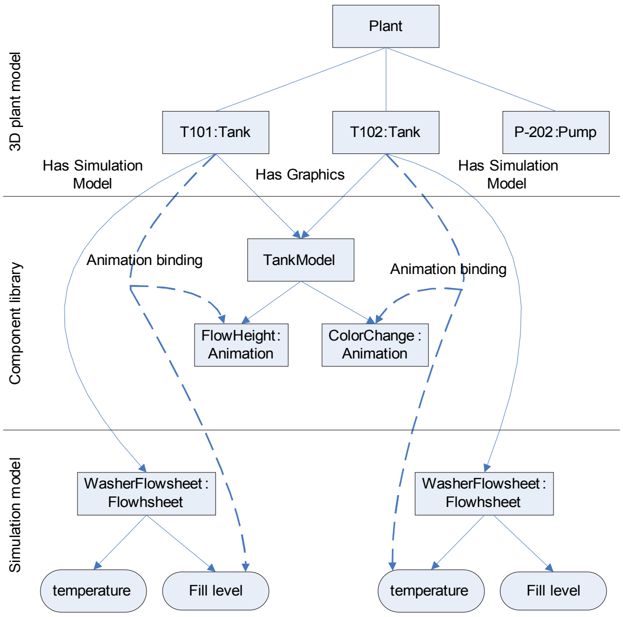

Figure 2. Overall data model. Model is divided to 3D plant model, component library containing animatable models of equipment, and simulation model.

5. Implementation Environment

Simantics is ontology based modelling and simulation platform developed (and is still under development) by Semantics Models research team at VTT. Design principles and history of the platform can be found in [16]. One of the key concepts of Simantics platform is separation of modelling data, simulation algorithms, and real-time data access. With this separation, the platform is capable of supporting multiple different models and multiple different simulation algorithms, with their own data structures, at the same time.

Simantics platform is multi-user modelling environment, hence it consist of two components: client application and server (Figure 3). Server, called ProCore, is decentralized, versioning triple storage, where all client applications are connected. Client application, called ProConf, is built on top of Eclipse RCP (Rich Client Platform) [16].

To support modelling, Simantics uses semantic graph, and its base ontology is previously described Layer0. It also contains set of foundation ontologies that provide basic concepts required in simulation and modelling, but were separated from Layer0. These include math, equations, units, structural modelling, and mapping. Math-ontology contains currently only complex numbers, Units-ontology contains units and conversion functions between them, and it allows user interface to use any units that user wants to use without breaking editors or simulators. Equations-ontology is extensible mechanism for handling mathematical equations. For instance, it is used in 3D modelling for parameterizing equipment models; A tank may have diameter and length properties that user can change and geometric representation of the tank is changed accordingly. Structural modelling ontology provides mechanism for reusing instance data of models, while providing unique property values for re-used parts. It is critical part for reducing required storage space, especially with simulation models, which typically contain complex property definitions.

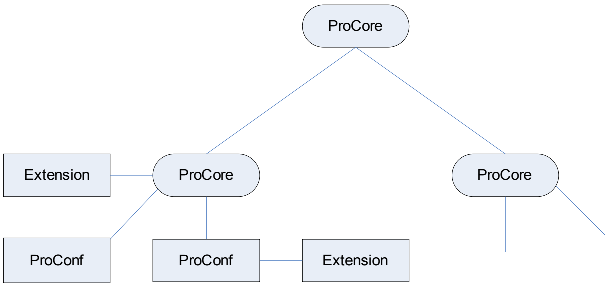

Figure 3. Architecture of Simantics Platform

Eclipse, which is used for user application, is plug-in based; core components contain only OSGi-based plug-in managers, which are capable of loading and unloading plug-ins when needed. Whole Eclipse-based application is just a set of plug-ins, which use each other's services to provide functionality that user needs. Plug-ins can use services of other plug-ins two ways: either extending other plug-in and its classes, or adding new functionality to existing plug-in using Extension / Extension Point mechanism. For ontology based programming Eclipse's plug-in architecture suits perfectly because the editors related to an ontology can be loaded only if the ontology is used, which decreases memory footprint of the program, when the user needs only few of the all available ontologies to do his tasks. An other way, how Eclipse's plug-in architecture is used, is that when an ontology extends concepts of another ontology, it can provide code that can handle those extended concepts by using the Extension Point mechanism.

For simulators, platform provides Extension mechanism. Extensions are client applications to ProCore, similarly as ProConf, but they are not tied to Eclipse and do not contain user interface. The most common use for the extension mechanism is simulators; extensions can be either registered to ProCore and it is able to launch them when needed, or extensions can be attached as part of local ProConf installation. Simulation data transfer between extensions, ProCore, and ProConf is done with Value Sets, real-time data access interface of Simantics platform. Value Sets are separated from triple database, and therefore are capable of transferring much larger amount of data.

Since ProConf is used for simulation, data visualization capabilities are one of required features of such platform. ProConf contains data visualization package called Webmon (Web technology based process data monitoring tool). Webmon is extensible visualization package, where new visualizations and data connectivity capabilities can be added as plug-ins. Currently implemented visualizations are 2D trend, binary signal, 2D curve, 3D trend, 3D surface, and 4D surface (Figure 4). Webmon can retrieve data from OPC XML-DA servers [17], import and export ASCII-based files, typical in automation systems, import data from and export data to historian server. Similarly capability to read data from Simantics' ValueSets has been added to Webmon, which allows it to visualize any simulation that is run in Simantics environment.

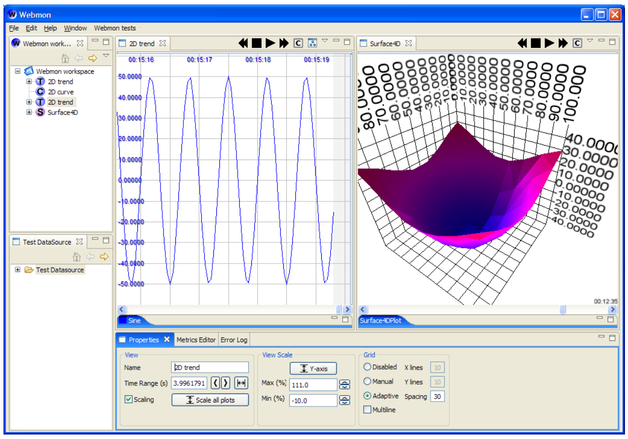

Figure 4. Data visualization component of Simantics.

6. An Example Case: Bleaching Line

As a practical use case, Process Editor is used for modelling a bleaching line that is part of a pulp mill. Own ontology is developed for this case. Ontology contains equipment that is needed to model the process (bleaching tower and washer) and mappings that generated simulation model from 3D plant model. Simulation model itself is divided to multiple ontologies, one describing basic flowsheet, one describing concepts of multiphase chemistry, and one specific to this simulation case, dubbed as Vista ontology. Name Vista comes from the name of project where multiphase chemistry simulation algorithms were developed [18].

Before mapping between 3D plant model and simulation model could be created, simulation models had to be linked to equipment models. While simulation model of a bleacher were implemented with a custom, algorithm [19], simulation of a washer was handled with more generic model. Therefore its simulation model had to be modelled with diagramming tools (Figure 5) before ontology for the test case could be created. The actual simulation model is generated by cloning flowsheets attached to equipment, and then linking ports in flowsheets to nozzles when the nozzles are connected to each other with pipelines. As a summary, one piece of equipment is always mapped to one flowsheet, and one pipeline to one flow.

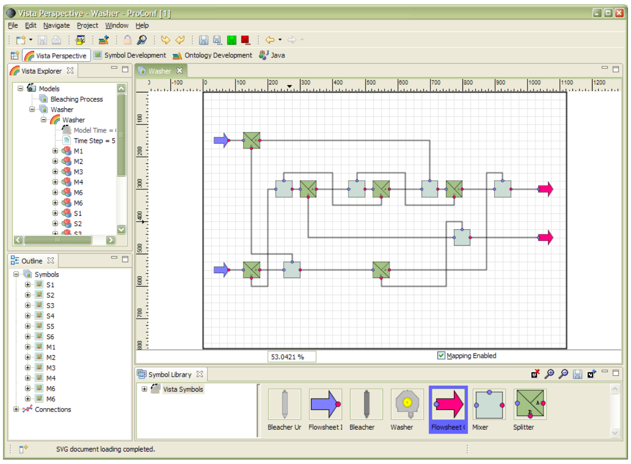

Figure 5. Multiphase chemistry model of a washer modelled with ProConf's diagramming tools [20].

Figure 6 shows visualization of the bleaching line, where particle effects are used for visualizing flows in pipes: colour of particles show cleanliness of the pulp and velocity of the particles is mapped to mass flow in the pipes. Effect of the first reactor can bee seen right from the image: particles are red when they enter the reactor and particles coming out of the reactor are violet. Difference of mass flow rates is much harder to see, even when pulp's mass flow is 190 Kg/s and water's mass flow is 295 Kg/s.

The problem of particle effects shows in the figure; particles are very hard to see at large distances. Another problem, which cannot be seen in the figure, is frame rate dependency of the particle effects: When computer cannot render visualization fast enough, not only seeing velocity of mass flow is hard, but also seeing direction of mass flow is impossible. This problem was hit when visualization was used in a laptop that used power saving mode, and its graphics processing capability was significantly reduced.

While particle effects give rather vague information about the simulation, text based monitors, which appear in the figure, can show precise numerical data. When the user enables monitors, simulation data from equipment that user is pointing is shown. Monitors can be configured, and here it is set to show mass flow rate, pH, and Kappa value of flows.

Test case showed that implemented animation features are not enough for all cases. The simulation model of reactors can provide Kappa in multiple points of the reactor. Simple colour change animation can change the colour of the whole object, and cannot create a colour gradient over the surface of model. Now Webmon's 2D curve and 3D trend plots were the only way to visualize kappa value's change inside reactors.

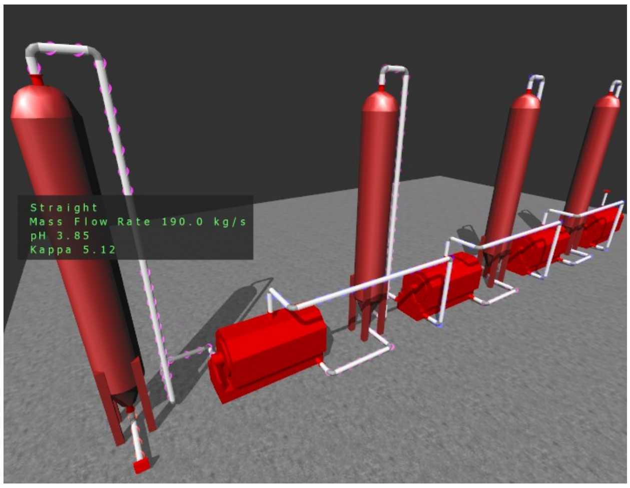

Figure 6. 3D model of a bleaching line with visualization of simulation results.

7. Future work

One of the main challenges for ontology based simulation environment in the future is scalability. As the example case of bleaching line shows, ontology based approach creates large amount of triples even for simple models. The future vision for Simantics platform is to use it as a user interface for VTT's own process simulation solvers, Apros and Balas. This sets a challenging goal for scalability: currently Apros has been used for simulating nuclear and conventional power plants and paper and pulp mills. These models may contain hundreds, if not thousands of equipment, and many kilometres of piping. When all that is described in ontological format, database will contain several hundred million triples. Therefore the example test case is just a tiny fraction of what the system should be capable of handling.

Work for this has already been started. Unnecessary relations have been removed from Layer0, and that reduces amount of triples needed by the models. Caching algorithms of ProConf have been re-implemented: Now it contains only one cache and it can also release cached triples. There has also been discussion of packing properties; instead of storing all relations, structured property is stored using arrays. This would reduce amount of triples especially with simulation models. Also clustering capability for ProCore has been implemented. This allows distribution of triples to multiple servers, reducing workload of one server.

If the scalability issue for triples can be solved, the next limit will be graphics. Designed caches of ProConf will allow it to handle larger models that will fit to the computer's memory. When a whole industrial plant is modelled, that is also required from graphical editors. When user is modelling the plant, it is more practical to filter unnecessary objects from the view, but when whole plant level simulation is visualized, the case is bit different. It depends on the user and what he wants to do: he may want to simulate the whole plant but is interested in only one section of it. Then similar filtering can be used, but when the user wants to just browse the plant and see everything, dynamic loading and unloading of graphical models is required.

8. Conclusions

The industry uses process simulators e.g. for process and automation design, control system testing and for training and support of operators.. Modelling and designing applications are currently separate from the simulation applications, and that leads to manual input of configuration data to simulator. Combining the design application and simulators into a single package provides means to tackle this problem. The designer can use simulators without extra effort and without information loss in the translation process.

3D industrial plant modelling has been widely accepted by the industry, and most of the new plants are designed with some 3D modelling software. When compared to traditional 2D process simulation tools, 3D graphics provide more lifelike view of the plant that benefits designers and end users.

We studied how 3D industrial plant modelling, process simulation and visualization could be combined with ontologies. Ontology-based approach was proven useful mechanism to break coupling of the user interface and the actual simulator. The visual model is separated from the simulator model, but since both models are in the same database, simulation can be configured through the 3D model. Therefore different simulators can be easily added without changing anything in 3D modelling and visualization.

Used visualization techniques provide means to better understand the behaviour of a simulated process. When compared to traditional text based monitors and graphical trends, animations are more intuitive and faster to interpret. When more precise information is needed, older methods are better; animations should not be seen as replacement, but as additional way to help the designer.

Easy extendibility comes at its cost: semantic triple model takes large amount of space, and requires lot of caching to work fast. This creates huge memory requirements and scalability of the method is an issue that represents a future challenge for the whole Simantics platform.

References

[1] Gruber, T. R. 1995. Toward principles for the design of ontologies used for knowledge sharing. Int. J. Hum.-Comput. Stud. 43, 5-6 (Dec. 1995), pp. 907-928.

[2] Uschold, M. and Gruninger, M. 1996. Ontologies: principles, methods, and applications, In Knowledge Engineering Review, 11(2), 1996, pp. 93-155.

[3] Guarino, N. 1998. Formal Ontology and Information Systems. Proceedings of the 1st International Conference June 6-8, 1998, Trento, Italy. 1st. IOS Press. pp. 3-15.

[4] Kalogerakis, E., Christodoulakis, S., Moumoutzis, N., 2006. Coupling Ontologies with Graphics Content for Knowledge Driven Visualization. Proceedings of the IEEE Virtual Reality Conference 2006 (IEEE VR '06), pp.43-50, Virginia, USA, 25-28 March 2006

[5] Park, M., 2005. Ontology-based Customizable 3D Modeling for Simulation. PhD Thesis. University of Florida.

[6] Lehtonen, T. and Karhela, T. 2006. Ontology approach for building and visualising process simulation models using 2d vector graphics. In Esko Juuso, editor, SIMS 2006 Proceedings of the 47th Conference on Simulation and Modeling, pages 141–146. Finnish Society of Automation, SIMS — Scandinavian Simulation Society, September 2006.

[7] Rohrer, M. W. 2000. Seeing is believing: the importance of visualization in manufacturing simulation. In Proceedings of the 32nd Conference on Winter Simulation (Orlando, Florida, December 10 - 13, 2000). Winter Simulation Conference. Society for Computer Simulation International, San Diego, CA, 1211-1216.

[8] Quick, J. M., Zhu, C., Wang, H., Song, M., and Müller-Wittig, W. 2004. Building a virtual factory. In Proceedings of the 2nd international Conference on Computer Graphics and interactive Techniques in Australasia and South East Asia (Singapore, June 15 - 18, 2004). S. N. Spencer, Ed. GRAPHITE '04. ACM Press, New York, NY, 199-203.

[9] McGuinness, D.L. and van Harmelen, F. 2004. OWL Web Ontology Language Overview. W3C Recommendation 10 February 2004. [Cited 15 Apr 2007]. Available at: http://www.w3.org/TR/owl-features/

[10] X3D Specifications. [Cited 10 Mar 2007]. Available at: http://www.web3d.org/x3d/specifications/

[11] ISO IS 15926-2:2003: Industrial automation systems and integration – Integration of life-cycle data for oil and gas production facilities – Part2: Data Model, 2003. International Organization for Standardization, Geneva, Switzerland.

[12] ISO IS 15926-4:2004: Industrial automation systems and integration – Integration of life-cycle data for oil and gas production facilities – Part4: Initial Reference Data, 2004. International Organization for Standardization, Geneva, Switzerland.

[13] Juslin, K., Paljakka, M. 1997. APROS - A Multifunctional Modelling Environment. Proceedings of the 2nd CSNI Specialist Meeting on Simulators and Plant Analysers, Espoo, Finland.

[14] Villberg, A. 2007. Design Challenges of an Ontology-Based Modeling and Simulation Environment. Master's thesis, Helsinki University of Technology, To be published in 2007.

[15] Hansen, C.D., Johnson, C.R., 2005. The visualization handbook. Elsevier, Burlington, MA. 962 pages. ISBN 0-12-387582-X

[16] Eclipse – an open development platform. [Cited 10 Mar 2007]. Available at: http://www.eclipse.org/

[17] OPC Foundation [Cited 10 Mar 2007]. Available at: http://www.opcfoundation.org/

[18] Räsänen, E., 2003. Modelling ion exchange and flow in pulp suspensions. PhD (Tech.) Thesis. VTT Processes. Espoo. 62 p. + app. 111p. VTT Publications : 495.

[19] Brink, A. Lindberg, D., Hupa, M., Louhenkilpi, S., Wang, S., Fabritius, T., Riipi, J., Härkki, J., Kangas, P., Koukkari, P., Lilja, R., Pajarre, R., Penttilä K., Kankkunen A., Järvinen, M., Fogelholm, C-J., Bergström, F., Eriksson, K. 2007. Multi-phase Chemistry in Process Simulation MASIT04 (VISTA). In Eija Alakangas and Pekka Taskinen, editors, MASI Technology Programme 2005-2009, Yearbook 2007, pages 35-46. Technology Review 207/2007. Tekes. ISSN 1239-758X, ISBN 978-952-457-367-2. Available at http://akseli.tekes.fi/opencms/opencms/OhjelmaPortaali/ohjemat/MASI/fi/Dokumenttiarkisto/Viestinta_ja_aktivointi/Julkaisut/Yearbook__2007_final_www.pdf [Cited 10 Sep 2007]

[20] Lehtonen, T. 2007. Ontology-Based Diagram Methods in Process Modelling and Simulation. Master's thesis, Helsinki University of Technology, May 2007.