



HAL
open science

Experimenting New Metaphors for PDM through a Model Driven Engineering Scheme

Frédéric Noël, Mohd Azwan Azli

► **To cite this version:**

Frédéric Noël, Mohd Azwan Azli. Experimenting New Metaphors for PDM through a Model Driven Engineering Scheme. 10th Product Lifecycle Management for Society (PLM), Jul 2013, Nantes, France. pp.570-583, 10.1007/978-3-642-41501-2_57. hal-01461907

HAL Id: hal-01461907

<https://inria.hal.science/hal-01461907v1>

Submitted on 8 Feb 2017

HAL is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L'archive ouverte pluridisciplinaire **HAL**, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d'enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.



Distributed under a Creative Commons Attribution 4.0 International License

Experimenting new metaphors for PDM through a Model Driven Engineering scheme

Frédéric Noël, Mohd Azwan Azli

Univ. Grenoble Alpes, G-SCOP, France

frederic.noel@grenoble-inp.fr, mohdazwan.azli@gmail.com

<http://www.g-scop.grenoble-inp.fr>

Abstract. Manufacturing companies invested a lot in Product Life Management (PLM) system to save their competitiveness by a better management of their internal processes. The most deployed systems are Product Data Management tools. By deploying such systems they try to formalize complex behaviour of the company. But PLM systems did not solve complexity and new gaps must be managed. At this point, it may be interesting to investigate new ways to overpass the new complexity gap. On another hand Virtual Reality, Augmented Reality and every other Visualization and Interaction technologies are investigating more intuitive modality to interact with information, but these technologies remain quite unused for PDM applications. This paper proposes the Model Driven Engineering horizontal transformation approach to provide a systematic way to explore Virtual Reality opportunities within the PDM scope.

Keywords: PLM, PDM, Visualisation, Interaction, Complexity, Virtual Reality, Model Driven Engineering, Intuitiveness

1 Introduction

The WP5.1 IFIP working group manifesto notes [1] that "Understanding the whole life-cycle impact of products is a critical issue in product development today". When the product development remains quite simple there is no rationale to deploy PLM. But indeed the increasing complexity of product and processes is a major trigger of technical evolution of PLM issues within industry. CAD system as many other simulation tools were a first answer to accelerate expert tasks and was a very good answer to support individual activity. But indeed as soon as more collaboration is expected, sharing information between experts became a new challenge. This step towards complexity was passed in the 80s and first Product Data Management systems appeared to answer this basic issue. It was first deployed at the service/team level where every expert has a good understanding of the colleagues activity. But complexity increased once again with both the necessity to share information at the company level, or between various stakeholders. Simultaneously, the number of actors and the number of expertises are increasing.

PLM claims to provide an holistic view, and thus, to pass over this last complexity step. But complexity of product life management remains a challenge. Product data Management tools (PDM) are the main component of the PLM toolbox: PDM appears as not so intuitive; sometimes as complex as the problem they are expected to solve. This paper makes the assumption that every information is at the end processed by users through various metaphors (indeed mainly graphical metaphors). The complexity appearance is thus included in the proposed metaphors and may be reduced by the exploration of new visualization modalities. The aim is to propose some solutions to decrease complexity of PDM solutions by using some virtual reality opportunities.

The paper focusses first on an overall Product Data Management (PDM) description to highlight its main concepts and functions. Section 3 presents the main added value of virtual reality technologies. Virtual reality (VR) is a promising technique for many issues. It has several successful developments for the interaction with a Digital Mock Up. At last but not least, Section 4 explores how model driven engineering (MDE)¹ may provide a systematic approach to create a link between VR solutions and PLM/PDM issues. It proposes a framework where, at a first step, PDM would take benefit from virtual reality. Indeed virtual reality may provide more intuitive and natural interaction with PDM information system. It may be almost a partial solution to pass over some complexity issues.

2 Apparent complexity of PDM systems

2.1 Concepts of Product Data Management

Following Pal&Betz model [2] of a product manufacturing company, the product life-cycle has a main sequential stream of activities including "concept creation, creativity", "Embodiment design", "Manufacturing", "Usage and Maintenance", "Disassembly" and "Recycling". This is managed by an overall strategy and must integrate the design of the product life. Let focus on PDM systems. These systems are characteristic of complexity management since they were the first attempt to organize collaboration within a design team, providing space to share information. In the next subsections the main concepts used within PDM systems and the main functions are reminded.

A PDM system mainly provides the following concepts when looking after product definition :

An article is the major concept handled by a PDM system. It organises the product structure tree shared by every contributors to design. The tree of articles is referred as the Bill Of Material (BOM). It usually corresponds to an assembly or a single component of the product but some companies also use it as a function tree decomposition.

¹ The paper uses the horizontal translation of MDE theory and does not practise vertical translation of Model Driven Approach (MDA) (translation from PIM to PSM)

- A document** is a logical set of information, to identify a usual computer document (text, spreadsheet, presentation, image, etc.). The PDM document refers one or several computer files and record some meta-data (author, creation date, version, validation state, maturity, etc.). Some documents have special associated attributes or function: CAD models provide a good connection between CAD software and PDM. When the document is uploaded from the CAD software a 3D standard view may be created.
- A directory** is an object where designer can pack several data together. A directory may contain documents or sub-directories as on a usual computer disk, but also articles. Sub-directories create a new hierarchical view of data. Companies can decide to maintain several hierarchical view of data by sorting them respect to services (CAD service, Simulation, Manufacturing process, etc.), or to business project, or any other organization. Multi-view introduces some complexity within the analysis of the graph of information since several trees provide access to information.
- A configuration** is a set of article versions that identify a specific arrangement of the product. Then in the same data-base several configurations of a same product are managed. The configuration is thus a snapshot of the product definition. Several snapshots can share common parts and identify some items which are different from one configuration to another one.
- A change Management item** is defined to ensure the management of changes. This is a specific process where the company must capitalize problem reports, modification demands, and the management of the change process. These objects obviously refer the previous ones. Indeed a problem may occur on a specific configuration of an article and must lead to a new version of a document.

More complex meta-model could be identified if we enter into project management considerations where team, user, role, forums, authority, organisation, tasks, deliverables, milestones, etc. should be linked to this model: but the product data mainly refers about five core concepts. Figure 1 organises these concepts within a UML diagram to provide a basic vision of their relationships.

The previous concepts are tangible for end-users: at this level of analysis, a PDM system should be very simple to use by any individual person with a few professional skill. The 5 previous concepts and their relationships lead indeed to complicate models. An PDM instance is a graph of concepts instances linked by relations compatible with a UML diagram (Figure 1 is such a UML diagram). This graph is rapidly not any more understood by a human brain and appears as really complex to manage. Learning PDM usage remains non natural and practise let experts concentrate on few function to avoid the overall complexity. Next section complete this PDM description with its main functions.

2.2 Functions for Product Data Management

The main functions for PDM systems can be listed as follow:

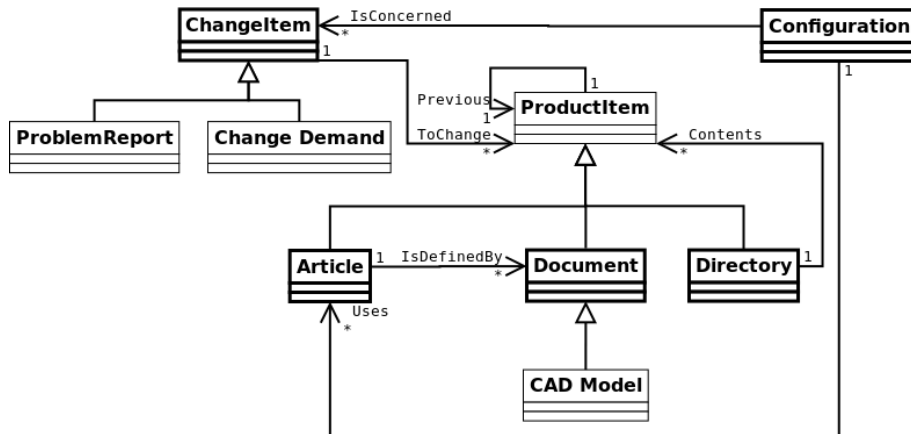


Fig. 1. A UML class diagram description of the main PDM concepts.

Creation of information: these functions allow to create new parts, documents, directories, configurations and change items. Whenever a model is updated, a new version is created through a check-in process.

Right Permission Management: Most PDM systems manage the concepts of User, Group or Role which enable data protection but also work-flow management. Depending on its state, (draft, validated, etc) every item can be accessed in read or write mode by specific users or roles. When a data is shared by several users, each one can reserve the modification to himself through the check-out mechanism. Indeed processes to merge concurrent edited version of a same document [3] were poorly studied within engineering models. The reservation of data which is released by the check-in mechanism is a way to short-cut the solution.

Navigation / Search: A key feature within every PDM system helps to find information. This may be done through navigation or global search mechanisms. Indeed these functions are more complex by the existence of versions and configurations. The multi-view property for every data usually provides search output with a list of documents which looks equivalent but are indeed small variation of the same item.

Work-flow : To support work-flow management is also a real advantage of PDM systems. The main work-flow is dedicated to change management. A second one was more lately integrated with project management issues. The work-flow is another multi-view dimension which makes PDM a complex system.

Complementary functions: Many other functions are provided by PDM systems but are secondary functions:

Document pre-viewer: some specific viewers/editors can be integrated to ease the perception of the document content. Usually CAD geometric models are well supported with pre-visualization of 3D snapshot. The

end-user has an overview and navigates around the object without edition facility.

Connectors: To ease interoperability with expertise tools connectors are proposed with specific software. CAD connectors are the most common practise and offer functions to directly access the PDM CAD documents from the CAD software. Check-in and check-out functions can also be integrated. At last but not least import of CAD models are available to create 3D documents or to identify the Bill of Material which is implicitly defined within the CAD system.

These functions are the main use case that can be applied by the end-user. But indeed the number of options and views that are proposed by PDM system let most end-users confused with the result of their action. Once again as for concept description, the PDM system appear a a complex system.

3 Virtual reality for PDM

Every activity requires rendering information. Newest visualization technologies are supposed to increase the efficiency of activities. However new visualization technique does not necessarily mean a better efficiency. This article aims to explore the opportunities opened by visualization and interaction for PDM activities.

3.1 Definition of Virtual Reality

In [4] Virtual Reality is defined as *"a technical and scientific domain using computer science and behaviour interfaces for the simulation of 3D artefacts behaviour within a virtual world. The 3D artefacts interact in real time with themselves and one or more users immersed through more or less natural sensor-engine channels"*. This definition clearly highlights the concepts which are handled through virtual reality. From a systemic point of view, the virtual reality systems is defined by:

its internal components: it may be artefacts of the virtual world. The geometric model of artefacts is a key point to identify artefacts of this world. But human user also participate to the virtual reality system. On pure virtual reality, users senses are disconnected from the real world unless mixed or augmented reality is referred.

its behaviours model the reaction of artefacts to events coming from the virtual and real world (captured with sensors) and may send feedbacks to the real world through actuators.

its functions: virtual reality enables real time interaction. That means that behaviours must be simulated in real-time or pre-simulated. The final user will accept to use virtual reality if it enables to achieve a task on a natural and efficient way.

To enable Virtual Reality usage for any use case (including PDM) expects to define the artefacts of the virtual world, the users who can be involved, their expected behaviours and a natural and efficient interaction model.

3.2 Variety of tools and techniques

Immersion techniques from virtual reality mainly call to mind 3D caves or head mounted devices but indeed the panel of technology is much more wide.

- In 3D caves stereoscopic views are projected on the walls of a room. The views are adapted to the tracked position of a single user.
- Head mounted devices provide a similar effect with two separated 2D displays in front of each eye.
- Holographic devices create a direct 3D perception (without glasses) which can be mixed with reality. Auto stereoscopic devices should provide the same effect on the user point of view even if technology is different with more limited 3D perception characteristics.
- Ultra-High definition (4K and 8K displays) exist in few institutes. However applications could increase more rapidly than immersion technologies. Indeed the installation of a high definition display consume less room than 3D caves. Standards are incoming and the deployment should be more rapid. Tiled displays is another way to extend the resolution and the size of the surface with respect to traditional displays [5]
- At last but not list displays on flexible material are under study and will open new opportunities[6].

Then the combinations of displays solutions make the virtual reality equipments quite complex to classify. Moreover displays are not the unique components participating to a virtual reality environment. It is clear that the perception of immersion depends on the technical environment [7]

Indeed the interaction mode is also fundamental. In most virtual cases the observer does not interact directly with virtual objects but through avatars. The avatar is handled by a deported device. The position of this visual avatar locates an indirect connection to a behaviour of the virtual scene. In few cases like with holography, the interaction with object may be direct. [8] classifies the various interaction devices respect to the following categories:

2D/3D devices: the device is considered as a 3D device whenever it manages the 3D position of an avatar.

number of degree of freedom (dof): tactile surfaces are usually plane and thus are 2 dof system even if we can imagine some 2.5 dof systems.

Haptic/Passive devices: Haptics provides feedback to the user by several metaphors.

At last some device are a combination of displays and interaction modalities. Multi-touch tables are a good example where a tactile surface is merged with the display surface.

This classification may help defining the tangibility of the virtual world [8].

3.3 Virtual Reality for holistic PDM

A few applications of virtual reality are already used for PDM activities: to navigate in a digital mock-up, to analyse the results of a scientific simulation, to apply ergonomics analysis, to extend collaboration by improving remote presence. These applications are dedicated to design and production but only few research projects are reported in the scientific literature with application for a more holistic view.

Graf et al. present several concepts to connect VR and PDM by using a software tool to navigate through the product structure using PDM information on VR [9]. The described concept covers the structure information but also all relevant PDM data and relationships between PDM objects such as articles, documents and projects. Involved designers should not be concerned about the data preparation, the creation of VR scenes in order to use VR. All product structures and part geometries in the two systems should therefore be kept up-to-date at any time. To achieve this objective, PDM system has to be a part of a PDM solution which manages all the information, required for the use of VR: it expects new metaphors for PDM.

Ralph H. Stelzer presents a process integration solution in which the VR components can be changed into basic collaboration interface of a Product Lifecycle Management (PLM) environment [10]. The connection between VR and PLM systems allow the designers to work in several VR sessions simultaneously on a product. Hence, PLM has a vital function to manage the data and to avoid conflicts during the collaboration work. This can be achieved by visualizing any information of the PLM system from a VR session. It is therefore possible to color the VR components as a function on their release status, highlight different structure allocations or view various modification states.

Hayka H. et al. present an integration solution to provide virtual reality applications with required data from heterogeneous environments such as PLM [11]. The solution uses the Gatekeeper, a Java-application that can be accessed through web service interface and communicates with other servers using web services, web protocols or ssh connections. The Gatekeeper collects data from various sources, secures the data provision and transfer and process submitted job in the background. A prototype shows that the Gatekeeper allows an easy usage of the results gathered during VR sessions and a PDM system independent usage of the data preparation methods.

Kim S. et al. present MEMPHIS or Middleware for Exchanging Machinery and Product Data in Highly Immersive Systems to centralize the communication between the CAD and the VR systems via a PDM system [12]. MEMPHIS main objective is to overcome the conversion issues between the CAD and the VR systems by managing the meta data of a product and other VR related property data through Meta Data Server. The VR data, which is needed to present 3D models of future products in real time and high quality, can be produced directly from PDM systems. Thus, it reduces the repeated correction integration process between the PLM data to VR data. ManuVAR is a project that develops a prototype combining PLM, virtual reality (VR) and augmented reality (AR)

and human factors (HF) methods [13]. It intends to integrate PLM and VR in a single environment.

All these studies expect a structured method to enable communication between PDM and VR. They emphasize the difficulty to manage this interoperability issue. Here it is also noted that immersion of virtual reality is mainly provided through the real perception of a third dimension not fully available with traditional 2D displays. Then the main interest of virtual reality for PLM should be to use this third dimension to simplify the understanding of a complex set of information. [14] defines visualization as the use of interactive and computerized visual representations of abstract data to improve cognition. The final user is the main purpose of visualization to help him managing complex set of information.

In [15] authors analyse directions and metaphors to simplify complexity analyses through visualization. It proposes a model driven engineering framework to support the automatic creation of extended views in a VSML (Visualization System Modelling Language). It would provide adapted metaphors depending on the user role but also on the device and focus interest. Three main views are identified:

A tree is a hierarchical representation. On an object oriented point of view, a tree is standardized with the definition of its root handle and for every item a method returning its childs. Each node and tree arc should be associated with attribute values that can be transformed into either, shapes, colors or pixmaps, or size, position or orientation.

A map is a reticular representation. A polyhedron visualizing a 3D shape is indeed a specific reticular representation, but every graph are included here.

A paysage, as data-scapes or data-mountains, localise information on a given surface. It expects the definition of the surface and the definition at specific location of highlighted issues.

A combination of previous items: every visualization item can be locally decomposed into a smaller metaphor: it enable to combine metaphors together. It can be also processed by a distribution over a tiled environment.

On a formal way every metaphor is parametrized by a finite set of parameters. To translate a data model into these metaphors should be driven by a limited set of mechanisms. Next section proposes the base of such mechanisms and thus enables 3D perception of PLM activities. The related mechanism based on Model Driven Engineering techniques will provide transformations as described in [15] and will enable the [15] proposal by providing effective mechanisms.

4 Transformation of PDM information into Visualization

4.1 Principle

Let suppose that we have a PDM model. Then a set of transformation rules must be defined to create potential visual metaphors and to distribute these metaphors other a set of renderers. The transformation must also identify behaviours and

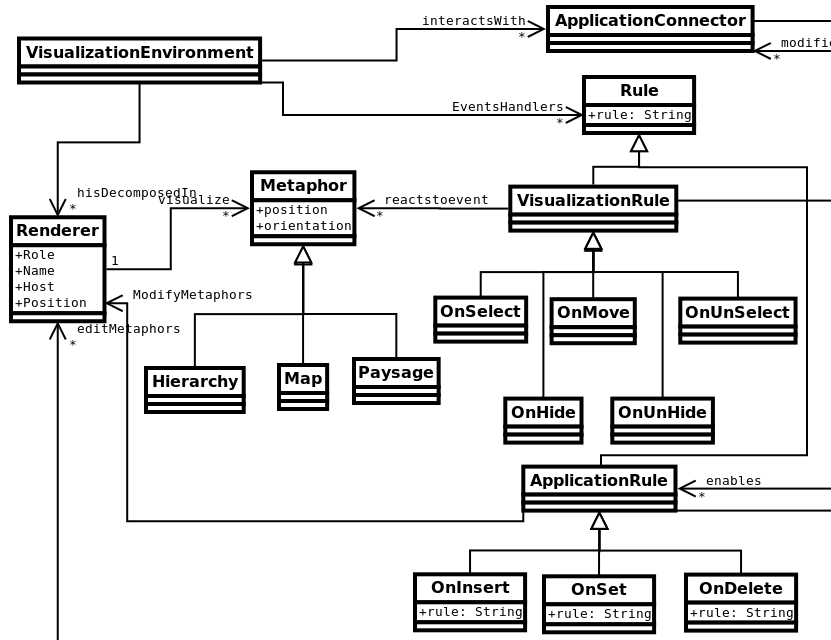


Fig. 2. Abstract description of the Visualization environment

will lead to interactive Virtual Reality framework. This principle has been implemented through autonomous agents. Agents communicate by sharing events on the common model of Figure 2. As stated by this UML diagram, the visualization environment is defined by a set of renderers. These renderers can be distributed on a network by identifying the host computer and their location within the computer rendered area. Every renderer is in charge to display a metaphor. The visualization environment also controls rules in charge to manage events coming from interaction with the renderer or from the visualized information system. To ensure the link with external applications we use an application connector which is in charge to survey in real-time the events produced by the application: here the PDM Model. This connector must implement the following API:

1. `GetRootHandle()` \mapsto [ID]: this method must return a list of unique identifiers of the application items. These "root" items will play the roles of the main handles on the model issued from the application point of view. For PDM this connector method should return for example a list of the main configurations. Thus user accesses a configuration of the product potentially decomposed into several alternative configurations. Each configuration provide access to product items, but indeed this can be adapted to special requirements depending on the PDM meta-model.
2. `GetItemClassifier(ID)` \mapsto CID : returns an identifier of the type of item identified by ID. For a specific ID the system should returns either 'Article',

- 'Configuration', 'Changed Demand', 'Problem Report', 'Document', 'Directory' or 'CAD Model'.
3. `GetClassifierAttributes(CID c1) ↦ [(name, type='Real'|'Integer'|'Boolean'|'String')]`: a specific item associated with the classifier `CID` has a list of attributes identified by their name and their type. This method returns the list of attributes for a given classifier. For every classifier a list of attributes must be associated. For an article or document the state 'Draft', 'Proposal', 'Validated' is one of these attributes. The cost could be a second one. The right permission as 'checked in' or 'checked out' should be another one, etc.
 4. `GetClassifierRelations(CID c1) ↦ [(name, card_{min}, card_{max}, RCID)]`: a specific item associated with the classifier `CID` has a list of relations with other classifiers. This method returns the list of tuples identifying the relations to a given `RCID` classifier. The relation is named and has a minimal and maximal cardinality. In the PDM case, as for attributes, every classifier has relations which are clearly identified for the PDM model on the class diagram 1. An article will have the 'IsDefined' relation associated to the classifier 'Document'.
 5. `Inherits(CID c11, CID c12) ↦ boolean` : This method return True if `c11` inherits from `c12`. For PDM, the class diagram 1 identifies inheritance relationships: the 'CAD Model' inherits the 'Document' class, etc.
 6. `GetItemAttribute(ID, attribute) ↦ Real|Integer|Boolean|String`: This method return the value of an attribute from a given item.
 7. `SetItemAttribute(ID, attribute, value)` : This method edits the value of an attribute from a given item.
 8. `GetItemRelation(ID, relation) ↦ [ID]` : This method returns the list of IDs from a relation of a given item.
 9. `InsertItem(ID, relation, IID)` : This method inserts a new item into a relation of a given item.
 10. `DeleteItem(ID, relation, DID)` : This method removes an item from a relation of a given item.

With this API the system read from and write to any application model. It enables to connect a PDM system and to visualize everything through various metaphors. The number of events in a PDM system are important since every collaborator should edit his work whenever expected. But indeed there is no streams of modifications that need rapid updates of the visualization scene. A check-in is done once and expects a check-out almost a few second later but not at every millisecond. That means that the visualization metaphors should support a Virtual Reality environment (no real-time issue). Mainly navigation is expected.

This API provides a generic way to navigate within both :

the meta-model level since the API provides information about the classifier, their attributes and relations (including inheritance).

the model level since it provides access to items, item attributes, and relations. The `SetItemAttribute`, `InsertItem` and `DeleteItem` functions allow edition of the model.

By managing the meta-model and the model level, this API opens access to a Model Driven Engineering approach as discussed in the next section.

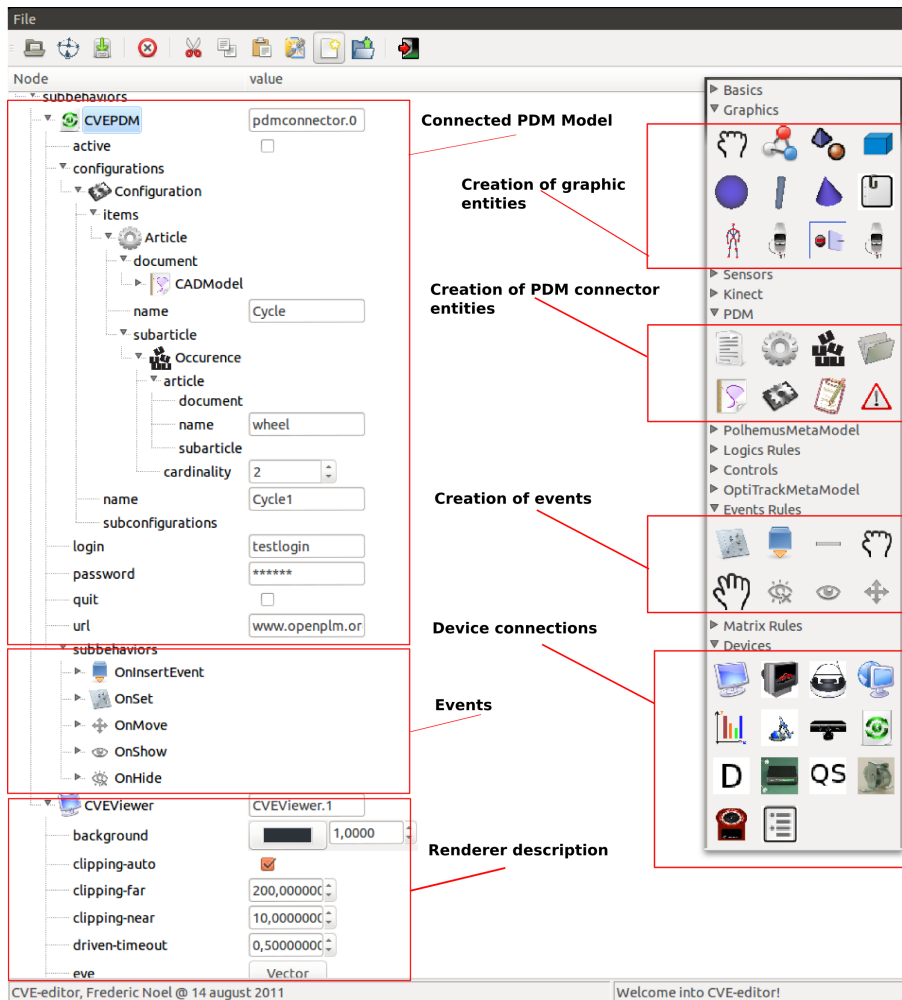


Fig. 3. Visualization Editor

4.2 Transformation rules

Model Driven Engineering (MDE) is based on the capacity to define transformation at the meta-model level. The current paper does not follow the usual model driven approach from computer engineering where translations are used on a vertical mode from an abstract model to a instance. This technique is usually

applied for the transformation of a Platform Independent Model (PIM) into a Platform Specific Model (PSM). This paper refers to horizontal translation and synchronization rules. The rules are defined at the abstract meta-model level and should be executed at the model level. The translation concerns two different meta-models: here from the PDM meta-model to the visualization meta-model. A major challenge for visualization in Virtual reality is to apply this translations in real-time.

Then to complete the interconnection between PDM and visualization, transformation rules must be implemented. Some rules are implicit as synchronization of attributes values or relationships between items whenever the visualization expect an edition or whenever the PDM system is edited, but more complex rules require a specific development. Especially the choice of metaphors must be identified and properly linked. The rules have also the ability to define behaviour of the virtual scene since rules can react to basic events (`OnSelect`, `OnUnSelect` or `OnMove`). Moreover the connector system can be extended to connect other applications in charge of simulating some behaviours (Mechanical effects as contact, etc within the scene).

To fix ideas an example of set of rules could be used: assuming that a 3D digital mock-up is available connected with the bill of material.

1. At initialization a list of product is visualized.
2. When a product is selected its digital mock-up is visualized as a usual 3D model.
3. When a part is selected, the list of documents must appear through specific 3D representations.
4. When a document is selected, the list of versions should appear on the line normal to the surface selected point.
5. When a version is moved towards a basket 3D model, the version must be deleted.
6. etc.

A prototype environment (named "CVE") has been developed for experimentation. The system acts through autonomous agent. Every renderer or device has its own autonomy. Connected agents can be of various type: management of interactive devices (tracking system) are also developped through specific agents. This environment enables to test various metaphors for any applications (including a PDM or other PLM application). Figure 3 shows the CVE editor which supports the configuration of the visualization environment. CVE demonstrates the powerness of the system and its capacity to project any application in various visualization metaphors on distributed renderers. Events, connector to the PDM system and graphic rendering are created directly through this editor.

At the meta-model level, transformation rules can be written by the VR tool developpers in a scripted language (in CVE currently python is used). But it can be also added directly by end-users through a basic scripting language accessing the models and the meta-models through the API of the previous section. This will ensure the creation of metaphors adapted to the real expected usage of end-users. The editor of Figure 3 also supports interactive description of rules. It is

a demonstration that visualisation rules could be defined by end-users with very few scripting capacity.

At the model level, when a new product is created in the PDM system (creation of an article at the highest level), a 'OnInsert' rule is called. This rule adds an object in the related visualization renderer but also defines the type of metaphor which must be used to visualize the PDM item. When the metaphor is selected in this renderer a 'OnSelect' rule is called: it invokes a new metaphor to display a focus on the product. This metaphor may replace the current view or be projected on any other renderer.

This environment demonstrates the capacity of Model Driven Engineering to support creation of metaphors for PDM activities enabling experimentation of new way to interact with PDM systems.

5 Conclusion

By organising visualisation as a systematic approach with a set of predefined metaphors, this paper opens new directions to visualize complex information set. It may support simplification of some views by creating unusual metaphors. 3D may be used to project things which were not thought as 3D and must not be reserved to 3D geometry visualization. The paper uses a core view of a PDM system; it proposes via a generic framework the PDM model visualization based on Model Driven Engineering transformations.

The capacity to use transformations towards Virtual Reality environments depends on the capacity of the transformations to be applied in real-time or to be prepared offline. The extension of mechanisms with interaction facility ensures a potential natural way to navigate within a virtual PDM view. The metaphors and related behaviours will have to be assessed in the context of PDM applications: the main issue is to compare intuitiveness and tangibility of every metaphor. It should lead to new metaphors for a better understanding and management of the apparent PDM complexity.

Acknowledgments. This paper was written within the scope of the VISION-AIR infrastructure project. VISIONAIR is leaded by Grenoble INP, 46 avenue Felix Viallet, F-38 031 Grenoble cedex 1, FRANCE. This project is funded by the European Commission under grant agreement 262044.

References

1. IFIP Working Group 5.1. Ifip working group 5.1 manifesto. <http://www.ifip-wg51.org/>.
2. G. Pahl and W. Betz. *Engineering Design: a Systematic Approach*. Springer Verlag., 1996.
3. Mohsen Sadeghi and Frédéric Noël and Khaled Hadj-Hamou. Formalization of design rules to assist conflict detection in collaborative design. *International Journal of Product Development*, 10(1/2/3):123–143, 2010.

4. B. Arnaldi, P. Fuchs, and J. Tissea. *Tité de la réalité virtuelle (chapter 1, vol1)*. Les presses de l'école des Mines de Paris, 2003.
5. Luc Renambot, Arun Rao, Rajvikram Singh, Byungil Jeong, Naveen Krishnaprasad, Venkatram Vishwanath, Vaidya Chandrasekhar, Nicholas Schwarz, Allan Spale, Charles Zhang, et al. Sage: the scalable adaptive graphics environment. In *Proceedings of WACE*, volume 9, pages 2004–09. Citeseer, 2004.
6. Gregory P. Crawford. *Flexible flat panel displays*. Wiley Series of Display Technologies, 2005.
7. W.W. Wits, F. Noël, and C. Masclet. Exploring the potential of 3d visualization techniques for usage in collaborative design. In *21st CIRP Design Conference*, pages 187–193, 2011.
8. Frédéric Noël. From integrated to intuitive design thanks to new technologies. In MITIP 2012, editor, *Modern Information Technology in the Innovation Processes of Industrial Enterprises*, pages 1–8. MITIP, 10 2012.
9. H. Graf, G. Brunetti, and A. Stork. A methodology supporting the preparation on 3d-cad data for design reviews in vr. In *Proceedings of the International Design Conference Design, Dubrovnik, Croatia*, 2002.
10. Ralph H. Stelzer. Virtual reality based engineering collaboration as part of the product lifecycle management. In *Proceedings of the ASME 2010 International Design Engineering Technical Conferences & Computers and Information in Engineering Conference. 2010, Montreal, Quebec, Canada*, 2010.
11. H. Hayka, D. Langenberg, R. Stark, and L. Wolter. Combining heterogeneous plm environments with grid computing and virtual reality applications. 2010. In *Proceedings of PLM2010 (Bremen Germany)*, 2010.
12. S. Kim and D. Weissmann. Middleware-based integration of multiple cad and pdm systems into virtual reality environment. *Computer-Aided Design and Applications*, 3(5):547–556, 2006.
13. D'Cruz M. Vink P. Krassi, B. Manuvar: a framework for improving manual work through virtual and augmented reality. In P. Vink and J. Kantola, editors, *Advances in Occupational, Social, and Organizational Ergonomics'(eds. Vink, P., Kantola, J.)*, CRC Press, ISBN 9781439835074, volume Proceedings of the 3rd International Conference on Applied Human Factors and Ergonomics (AHFE2010), Miami, Florida, USA, 17-20 July, 2010, 10 p. ISBN-13: 978-0-9796435-4-5. Published, 2010.
14. Ben Shneiderman Written and Edited By Stuart K. Card Jock D. Mackinlay. *Readings in information visualization: using vision to think*. San Francisco (Calif.) : M. Kaufmann publ., cop. 1999, 1999.
15. D. Bihanic and T. Polacsek. Visualisation de systèmes d'information complexes. une approche par « points de vue étendus ». *Studia Informatica Universalis.*, 10:235–262, 2012.