HAL open science

Neutral Interface for Assembly and Manufacturing Related Knowledge Exchange in Heterogeneous Design Environment

Minna Lanz, Roberto Rodriguez, Pasi Luostarinen, Reijo Tuokko

To cite this version:

Minna Lanz, Roberto Rodriguez, Pasi Luostarinen, Reijo Tuokko. Neutral Interface for Assembly and Manufacturing Related Knowledge Exchange in Heterogeneous Design Environment. 5th IFIP WG 5.5 International Precision Assembly Seminar (IPAS), Feb 2010, Chamonix, France. pp.21-29, 10.1007/978-3-642-11598-1_3 . hal-01055666

HAL Id: hal-01055666

https://inria.hal.science/hal-01055666v1

Submitted on 13 Aug 2014

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

Neutral Interface for Assembly and Manufacturing Related Knowledge Exchange in Heterogeneous Design Environment

Minna Lanz, Roberto Rodriguez, Pasi Luostarinen, Reijo Tuokko

Tampere University of Technology, Department of Production Engineering, P.O.Box 589, 33101 Tampere, Finland
{minna.lanz,roberto.rodriguez,pasi.luostarinen,reijo.tuokko}@tut.fi
http://www.tut.fi/tte

Abstract. The goals of this research are to provide overview of the recent activities in the field of design of manufacturing and assembly processes from the knowledge share point of view and propose a solution how to connect products to the manufacturing or assembly processes and suitable systems. The research conducted here starts with the assumption of connectivity of three design domains: product, process and system design domains. The characteristics of the product are pre-describing the set of the processes needed to manufacture and/or assemble the product. The description of the product defines constraints for the suitable processes. The processes are pre-defining the system requirements and constraining the set of systems capable of carrying out the needed processes. From the technical point of view the goal is also to provide an information architecture, where the relations exist between these three domains.

Key words: ontology, knowledge exchange, knowledge base

1 Introduction

Global economy has made manufacturing processes to become more and more distributed around the globe. Design teams are located possibly in every continent. In such a geographically and temporally divided environment, effective and profficient collaboration between design teams is crucial to maintain product quality, production effciency and organizational competency.

Partly due the globality the nature of manufacturing facilities has been in continuous change in past decades. The emergence of highly computerized design and control systems are changing the fundamental assumptions of how the production should be planned and controlled. Centralized systems based upon the need to share data by point of need acquisition are giving a way to global dristibuted production network enabling as well as requiring localized and fast adaptation to the production changes.

In the recent years industrial standards are being defined in a more computerreadable form, most notably since the emergence of XML-based formats and

computing power. XML as a language has a number of advantages for developers and implementers, because these specifications can be compiled by computers, databases can be automatically built, and certain kinds of testing can be performed more easily. However, some groups have used XML markups as a substitute for modeling the information - a dangerous shortcut that only works in communities that already share a common understanding of the meaning and usage of terms. Based on Ray [8] a far better approach is to adopt one of the emerging semantic technologies, such as Web Ontology Language (OWL), or first order logic.

2 Challenges from the industry

There are several barriers for achieving an integrated collaborative production facility. One of these is the challenge with the proprietary information representations, which often force the experts to serve as manual human-machine interfaces between otherwise automatable functions with the concomitant introduction of errors, ambiguity and misinterpretation. This leaves the systems unable to share the information with required speed and accuracy. Today the problem is somewhat solved, unfortunately with considerable expense coming from highly customized information mapping, translations and add-hoc implementations knit together forming unstable structure of mainly shallow (meaningless) information [8] [2].

This leads to several problems; firstly the communication between departments' "designer domains" becomes time consuming since the models have to be created over and over again.

The second problem is that by every remake and update the model actually loses information, because the second tier does not require all of the information created in the first phase. The result is that there are multiple sources contributing specific knowledge to several isolated models and revisions instead into "the master model" [2].

The third problem that comes next these two is the re-use of the knowledge. At the moment the re-use of the existing information is quite impossible due to the several models which are meaningless from the knowledge-point of view. For example the production planning knows that there are several stations, robots and grippers which can be re-used in the production of the next product family needing only a bit of modification. However, despite the amount of stored information, there are no up-to-date complete information set easily available of the interfaces and life-cycle-data and dimensions of the station and its components. In most of the cases the new station and/or the line is therefore designed starting with an empty layout [2].

The fourth problem which follows, is that the even the semiautomatic reasoning becomes impossible. The decision support systems are at best unreliable and relying heavily on the users' expert knowledge of the product, processes and resources. There is a lot of potential for the re-use of the existing knowledge of the models. But once again the problem is that there is no definition, the

ontology, which would explain what the system is, what its capabilities are and how long it has operated or why it was designed as it was in the first place. And even if the ontology would exist, the modern CAD/CAPP systems which are currently the market leaders cannot utilize the existing information [2].

The aim of the paper is to introduce a model for combining design information from several different sources under one reference model which can be accessed via common interface. The model itself utilizes feature-based modeling methodology as a core in order to combine product, process and system related information to the closed models of existing design tools.

For the product knowledge representation there exist the well known product knowledge sharing format such as Standard for the Exchange of Product Model Data (STEP) and it's extensions. For process and system descriptions there are also different standards addressing the needs from those domains, such as Process Specification Language (PSL) and Core Manufacturing Simulation Data (CMSD). However, these standards as well they are defined can be seen as "islands of standards" , since there are no models for representing all of these domains under one architecture [8]. In most of the implementations there are overlaps between standardized models but again they are not interconnected. The second note has to be made, that the standards have evolved over the years and become very complex set of extensions, dedicated to one view point and serving mainly that specified view point.

There exist different implementations of standards linked to each other, but very often these are forming a framework for expert systems, where the actual reasoning and design rationale is embedded into the model itself. The specified, yet detailed models are in most of the cases incomplete to be used as such in the case scenarios. The purpose of this work is to form a demonstration of the model where the different domains are indeed connected under one model in such manner that the model is general enough to be used in different case scenarios of which one is introduced in the Chapter Case.

3 Approach

3.1 Theoretical approach

Currently the knowledge is exchanged between two or more systems through dedicated interfaces. When the company's design tool platform includes up to hundred different design tools and software the point-to-point integration between all of the systems becomes quite impossible. In order to manage the information exchange between systems, the exchanged information is stripped and only the basic geometry or simplified data structures are exchanged. This leads to the situation where the design is done in strongly filtered snap-shot information. In order to address the process and system related information through product description and reason based on the existing information another approach must be adapted.

The right side of figure 1. illustrates other option for information exchange by utilizing common information model which is accessed through one interface.

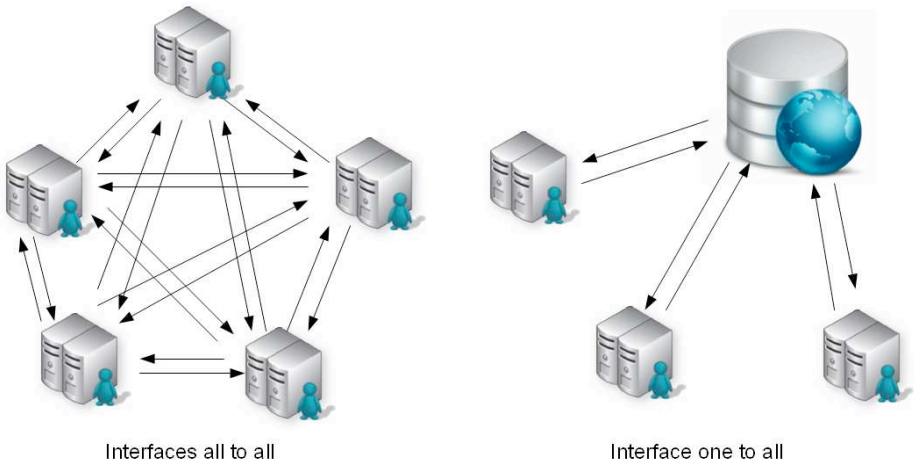

Fig. 1. Knowledge exchange between clients

In this case all of the overlapping design systems can contribute and retrieve their knowledge into a common knowledge model. In this case the focus is in contribution of knowledge rather than filtering and re-creating it. The clients here naturally cannot utilize fully all of the information but can use what they need and consider the remaining information as metadata. Metadata can be accessed and interpreted by humans as well.

The approach utilizes the theoretical methodology of product, process and assembly system connectivity graph introduced by Rampersad [7], who focused on the theoretical model of how the connectivity was shown in between of these three domains. Lohse [4] continued developing the integrated assembly model by concentrating to the connectivity of assembly processes and available assembly systems. The approach taken here concentrates the connection of products and processes as the starting point.

3.2 Implementation

The domain ontology for representing the connections, Core Ontology, illustrated in figure 3, was created with Protege 3.3.1 Owl editor and visualized with GraphViz. The Core Ontology consists of three domains for describing a content and context of a model. The product section of the ontology has four main levels *product, sub-assembly, part* and *feature*. The connection between products and processes is done in such way that the *Processes* are connected to the *Product* level, *Task* to *Sub-Assembly, Operations* to *Part*, and *Actions* to *Geometric Feature* level. The resource model consists of classes for devices, actors, tools, and area (having subclasses for factory, line and station) for describing the context of the manufacturing environment. The model allows the occurrence of multiple devices such as robot and drill combinations.

The Knowledge Base (KB) was designed to be a system where the data could be stored and retrieved for and by different applications varying from the product and process design to simulation. The first step in the development was to design an architecture that will allow the contribution and sharing of information between different applications. Figure 2 shows the planned architecture for the knowledge base. To achieve this design the following tools were used to facilitate the approach:

- Apache Tomcat web server,
- Apache 2 web services engine,
- Jena semantic web framework [1],
- Pellet reasoner [5], and
- Postgre database.

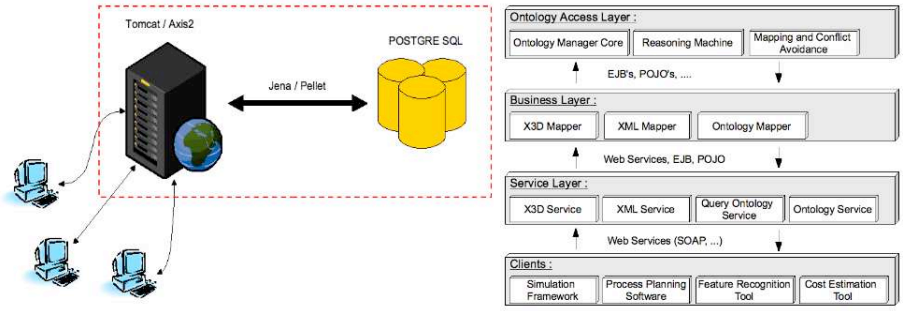

Fig. 2. Structure of the KB architecture and its implementation

One way to fulfill the challenges discussed in the second chapter is to use a standardized way to communicate with the clients. For that a Service Oriented Architecture (SOA) was chosen to be used in this implementation. SOA is defined by W3C [12] "*A set of components which can be invoked, and whose interface descriptions can be published and discovered*". SOA is a standardized architecture that can be implemented by using a set of Web Services, which are software systems designed to support interoperable machine-to-machine interaction over a network and an interface described in a machine-processable format eg. WSDL (Web Service Description Language). Other systems interact with the Web service in a manner prescribed by its description using SOAP-messages, typically conveyed using HTTP with an XML serialization in conjunction with other Web-related standards [10]. With the use of Web services the client applications could interact with the system by sending or retrieving information for the KB.

Originally, the architecture was designed to host different kind of services, depending on the needs of the applications. As result, the client applications were, indeed, the ones who pushed forward the development of the KB. The first

set of Web services was developed based on the classes defined in the Core Ontology, see figure 3. After a while, when the development of applications started, additional Web Services were needed to fulfill the needs of the client applications.

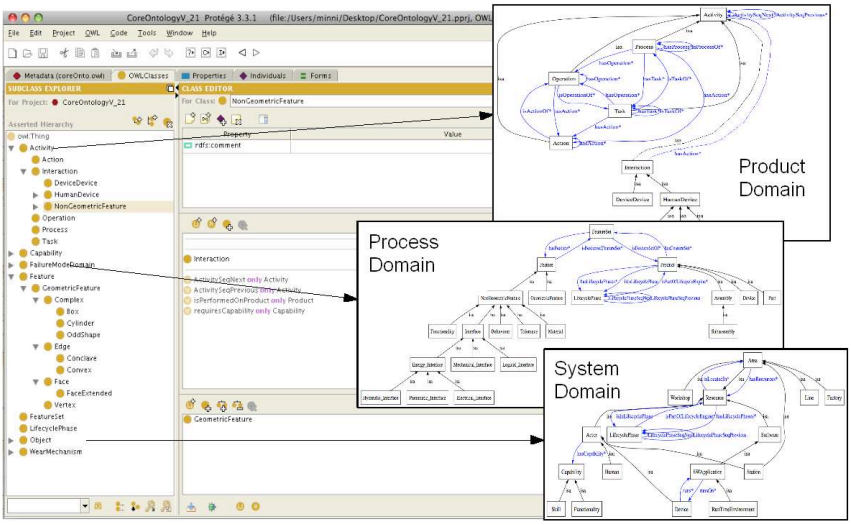

Fig. 3. CoreOntology for addressing Product, Process and System knowledge

The Web Services are segmented into five categories in order to simplify the development of the KB, but also to group Web services with similar functionality. This division is only logical, since all of them run on the same Web server and was done to improve the development of the code, since it is unmanageable to maintain and correct errors on the programming code when the number of services have grown. In addition, it simplifies the tasks of developers of client applications since it allows to target easily which services they would need for their development. The resulting categories of Web services are the following: product, process, system, model and general.

3.3 Reasoning

The reasoning in the KB is done in two different ways. First, the CoreOntology stored in the KB is founded in classes. Each of the classes has properties and relations to other classes. For example the class Product is related to the subclass Parts, since a Product has Parts. Every time new data is stored in the KB, new instances of those classes are created and where the new data is classified. Since the number of classes, properties and the relationships between them are static and not modifiable by the users, the KB can define what is the relation between classes. In this way fundamental reasoning is achieved.

The second way of reasoning is by using a reasoner inside the KB, the reasoner used in this implementation is Pellet. Pellet allows creation of rules to define extra relations between classes and complex queries to be done by calling a simple request [5]. At the moment there are few queries in the KB that use Pellet as a reasoner. Some extra queries are being done at the moment to exemplify the use of Pellet in the KB and how it could be used for real use cases.

3.4 Interfaces

Initially, the KB implementation had only two different kinds of services: Ontology Management Services and Ontology Querying Services. These services are in direct contact with the ontology and reside in a private part of KB where they cannot be accessed from the outside world. The majority of those services have an interface to the world through Web Services, which allows the KB shares information with other applications. Due the challenges of the clients' interfaces and lack of capability to understand new classes the functions to save and retrieve of proprietary/closed files from/to KB was added. The Ontology Querying services were divided in four main groups: Product, Process, System and Model Web Services depending on which part of the ontology they were designed to query.

This division into several groups helped to guide the developers to create different applications to consume the Web Services that focused in the part of the ontology they wanted to target. Furthermore, this grouping improved the ease of usability when referring to the documentation for developing client applications since it gives a quick information about what the application is targeting.

4 Case Study

The case study is inspecting and testing the developed ontologies as knowledge exchange media between the manufacturing and assembly design systems in heterogenous environment, while all of the the clients are utilizing propietary data structures. In this case the process plan of the product is first created in the process planning tool, CAMeLEAN. The planning of manufacturing/assembly processes with CAMeLEAN is dedicated to the processes and resource description in the shop floor. CAMeLEAN describes the processes in a workstation and focuses for balancing processes between the work-stations of a complete production line. The processes generate or manipulate parts of the final product or the product itself. Processes have cycle-times, predecessors and successors and depend on resources defined via this tool [11].

The goal of the case study is to inspecting the aspects of how to link product and process models and to use those (semi)automatically in the simulation tool. The underlying idea is to utilize the ontological product and process definition to populate the simulation environment as automatically as possible. Currently the client, illustrated in the figure 4, built inside the Visual Component's 3DCreate simulation environment access straight to the KB client and queries whether or

not the product description has reference parts or visual components connected to the ontology. The process plan is simulated inside the 3DCreate and validated processing times are updated to the KB. Moreover, in case that changes there have been made on the layout or resources used the corrected models can be uploaded to the KB.

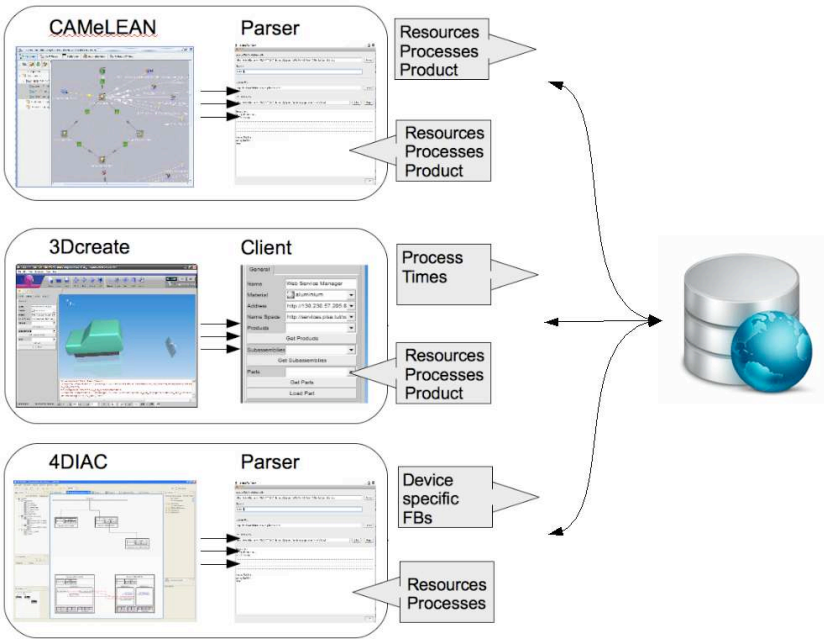

Fig. 4. Required knowledge exchange between the design tools

The Framework for Distributed Industrial Automation and Control (4DIAC) is being integrated with the PiSA SP3 tool chain for programming the various control devices within the production process. The control programs developed with the 4DIAC-IDE (Development Environment) are programmed with the IEC 61499 standard and uses Function Blocks as the main building blocks [9]. Within the PISA project, the 4DIAC-IDE is connected to the KB and stores Function Block Types (FBT) within the KB, as can be seen in the figure 4.

The interaction between CAMeLEAN and 4DIAC takes places via the KB. CAMeLEAN defines the preliminary process layouts and resource planning including the hardware components being used within the process. 4DIAC retrieves the processing times and resource information and adds the detailed representation of hardware elements and corresponding FPs to the KB.

The case study explored the defined integrated ontology from the knowledge contribution and share point of view. The clients here are used in more in the design phase than actual manufacturing. The clients are able to modify, add or

delete one or all of the instances concerning the particular product or process description.

Another case study was conducted where the same ontology and knowledge architecture was used for conveying the planned processes and corresponding product to the manufacturing environment. The CoreOntology was able to capture the design knowledge in digital, virtual and real knowledge domains. The digital represented the plans and design knowledge, virtual knowledge consisted of simulated processing times and real knowledge capture of the processing times and measured values during the manufacturing process. As addition to the case study represented above the controlling interface of the used device combination (considered as client) returned the operational values back to the KB. The product model was updated with real values and the simulation times for features were validated. This ontology also defined the basic communication between holonic entities inside the holonic manufacturing TUT's DiMS (Digital Manufacturing System) framework, for more information please see [3].

5 Conclusions

The approach introduced here aims to explore the possibilities of utilizing the product-process-system related information in an environment where different clients are contributing information in different levels of abstraction. The model consisting the product, process and system related knowledge was formed and tested in heterogenous design environment. The model was capable to include the contribution from different design tools under one generic knowledge representation. The current status allows the meaning of the content to be combined with proprietary/closed formats by offering references and content description for the models.

6 Future Work

The CoreOntology used at the moment is very generic by nature and the future work focuses on developing more detailed resource models to capture capabilities, geometric properties and other metadata relevant to that particular resource. The ontology needs to be improved in such manner that it can capture and convey the operational parameters of each resource into the simulation environment where the different production scenarios can be tested before applying into the factory floor. Further more, the ontology must be able to capture both the history of what happened to the product and the temporal description of each resource. By saving the resource's operational history, the backtracking of the systems' conditions, energy consumptions or parts' design features becomes more feasible. In addition, the tool's current use state and other machinery's characteristics can be saved to other KB and used for improving the process efficiency, pinpointing the possible quality problems related to the part's features or single machines and overall improvement of various processes.

Acknowledgments. The authors would like to thank 6th Framework Programme Integrated Project PISA - Flexible Assembly Systems through Workplace-Sharing and Time-Sharing Human-machine Cooperation consortium of their valuable feedback during the design and implementation of the system.

References

1. Jena `http://jena.sourceforge.net/`
2. Lanz,M., Kallela, T., Velez, G., Tuokko, R.: Product, Process and System Ontologies and Knowledge Base For Managing Knowledge Between Different Clients. IEEE SMC International Conference Distributd Human-Machine Systems, pp. 508–513 (2008)
3. Lanz, M., Tuokko, R.: Generic Reference Architecture for Digital, Virtual and Real presentations of manufacturing systems. Proceedings of Indo-US Workshop on Designing Sustainable Products, Services and Manufacturing Systems, `http://www.mel.nist.gov/publications/publications.cgi`,(2009)
4. N. Lohse, Towards an Ontology Framework for The Integrated Design of Modular Assembly Systems. PhD thesis, University of Nottingham, (2006)
5. Pellet `http://www.mindswap.org/2003/pellet/`
6. Rachuri,S., Han,Y-U., Foufou, S., Feng, S.C., Roy,U. Wang,F. Sriram, R.D., Lyons,K.W.: A Model for Capturing Product Assembly Information. Journal of Computing and Information Science in Engineering March 2006 vol6/11, (2006)
7. Rampersad, H.K.: Integrated and Simultaneous Design for Robotic Assembly. ISBN-10: 0471954667, (1994)
8. Ray, S.R.: Tackling the Semantic Interoperability of Modern Manufacturing Systems, Proceedings of the Second Semantic Technologies for eGov Conference, (2004)
9. Rodriguez,R., Lanz,M., Rooker, M.: Task 3.4.1. Implementation of the Knowledge Base. PiSA SP3 Task report. p.22 (2009)
10. Thomas, E.: Service-Oriented Architecture: A Field Guide to Integrating XML and Web Services. Prentice Hall, Upper Saddle River, NJ, ISBN:0131428985 (2004)
11. Velez Osuna,R.: SP3 Task 3.1.4 Conceptual Toolset Definition. PiSA SP3 task report 3.1.4. p.38, (2007)
12. World Wide Web Consortium, Service-Oriented Definition W3 Glossary, `URL: http://www.w3.org/2003/glossary/keyword/All/?keywords=service-oriented`