Proceedings of the ASME 2010 International Mechanical Engineering Congress & Exposition
IMECE2010
November 12-18, 2010, Vancouver, British Columbia, Canada

IMECE2010-37486

INTEGRATED PROCESS PLANNING BASED ON A FEDERATIVE FACTORY DATA MANAGEMENT

Christian Mosch
Technische Universität Darmstadt
Department of Computer Integrated Design
Darmstadt, Germany
mosch@dik.tu-darmstadt.de

Reiner Anderl
Technische Universität Darmstadt
Department of Computer Integrated Design
Darmstadt, Germany
anderl@dik.tu-darmstadt.de

Antonio Álvaro de Assis Moura
Methodist University of Piracicaba
Lab. for Computer Application in Design and Manufacturing
Santa Barbara d'Oeste, SP, Brazil
atamoura@unimep.br

Klaus Schuetzer
Methodist University of Piracicaba
Lab. for Computer Application in Design and Manufacturing
Santa Barbara d'Oeste, SP, Brazil
schuetzer@unimep.br

ABSTRACT

Manufacturing companies are confronted with short product life cycles, more variety of products and short cycles of leap innovations. This results in a higher frequency of changes in factory structures and an increasing importance of factory planning processes. Factory planning processes are characterized by participative and interdisciplinary processes due to various actors dealing in different domains and working in distributed environments. The result is a heterogeneous IT-landscape based on increasing use of multiple isolated and domain-specific IT tools and systems and hereby an increasing redundant, inhomogeneous and inconsistent data-holding. The control of these factory planning processes can be reached by holistic approaches and consistent system integration. The mean of system integration is the consideration of all domains involved in planning processes, used IT tools and systems and business processes. In this paper, the approach of a Federative Factory Data Management (FFDM) based on Service Oriented Architecture (SOA) and Semantic Model funded by the DFG (Germany) and CAPES (Brazil) will be described, which faces up the described challenges of factory planning processes. The focus of this approach is on the integration of isolated used IT tools for the dimensioning and structuring of factory systems, the generated domain-specific partial models as well as the coordination and synchronization of engineering workflows. In order of the control of factory planning processes the integration and coupling of the views of products, processes and resources on metadata level is required for a communication between different isolated and domain-specific IT tools of the various involved domains without losses or redundancies. The integration and coupling of these three views is based on a document independent factory structure description linked with factory defining metadata. In order to integrate and couple these different views, the relevant information and independencies are identified. Current reference process models for production and factory planning as well as the current methods to describe domain-specific models are analyzed. This is the basis for the development of the FFDM to build up a semantically coherent information model as a common communication and integration framework to represent the factory and to define and to access factory data. The goal of the presented approach is the increasing of planning harmonization, certainty, quality and frequency by a consistent information flow as well as the reduction of time of product development and factory planning processes.

INTRODUCTION

The existing conditions at the market and production environment exacerbate the product development and the factory planning and have an increasing of complexity of these processes as a result. The command of the complexity in product development [1, 2] as well as in factory planning [3, 4] is one of the key challenge. The command of complexity in factory planning is the focus in this paper. The factory planning process can be described as an evolutionary, participative and interdisciplinary process [5]. One way to handle the described processes are holistic approaches and consistent system integration. The meaning of system integration is the description of process chains as well as the integration of all participant domains, systems and models in the planning process. System integration got an indispensable process, which has the increasing interdisciplinary and usage of (most isolated) IT tools as a reason. The increasing usage of isolated IT tools results in deficits of data holding. The reason of this trend can be explained with the goal of using deterministic methods and the necessary IT tools within the digital factory. The integration of the isolated IT tools, the command of the complexity and the harmonization of planning processes is the goal of the federative factory data management (FFDM). The FFDM, based on service oriented architecture (SOA) and resource description framework (RDF), has to build up the basis for a harmonic data exchange between original equipment manufacturer (OEM) and suppliers on the one hand and to challenge the interoperability domain-specific IT tools on the other hand.

CURRENT SITUATION

Integration of the isolated IT tools and the command of the complexity and the harmonization of planning processes is not a new topic, both in product development and factory planning. Several IT tools or management systems for the integration of IT tools in product development and factory planning are available. Nevertheless, an improvement for the integration of both domains (product development and factory planning) is necessary. Management systems for the integration of IT tools in product development, called product data management (PDM) systems, focus on the management of product related data. Thus the data models are based on the requirements of product data management and not on factory data management. Hence the necessity of extending these data models with generic characteristics of a factory does exist [6]. The extended data model has to consider the management of product related data as well as the management of production related data [7]. As production related data process data describing the structure of factories and resource data describing the resources like machines for producing the product (product related data) can be mentioned.

PROBLEMS IN FACTORY PLANNING PROCESSES

In producing companies, several domains are involved in factory planning processes. For factory planning, information about the product, the resources in the production facility as well as the information in production processes is essential. Therefore domains like product development and production planning have to interact in an interdisciplinary way. These domains are working in different IT environments. Product development uses management concepts like PDM systems and tools like computer aided design (CAD) systems. The IT environment of production planning reaches from management concepts like enterprise resource planning (ERP) systems to tools like computer aided manufacturing (CAM) systems. Also the IT landscape within the factory planning processes is not such integrated as in product development [8].

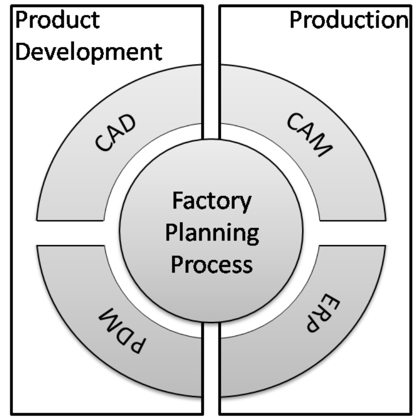

Figure 1: Factory planning process within different domains

The major problem in factory planning process is the integration of these various IT management concepts and IT tools as well as the synchronization of the engineering processes. Considering domain-specific workflows, every department involved in product development or production planning uses different isolated IT tools and systems generating domain-specific documents and data.

DOCUMENTS AND DATA

Following the difference between documents and data has to be specified. Referencing to Deutsches Institut für Normung e.V. (DIN) a document is defined as a unit of a handled summary of information [9]. Probst et al. [10] defines information as data combined with semantic. Data in turn is a result of signs following a specific syntax. Within the context of this paper, (factory defining-) data is in focus of consideration. For an interdisciplinary interaction of various domains, the exchange or provision of data is core of interest.

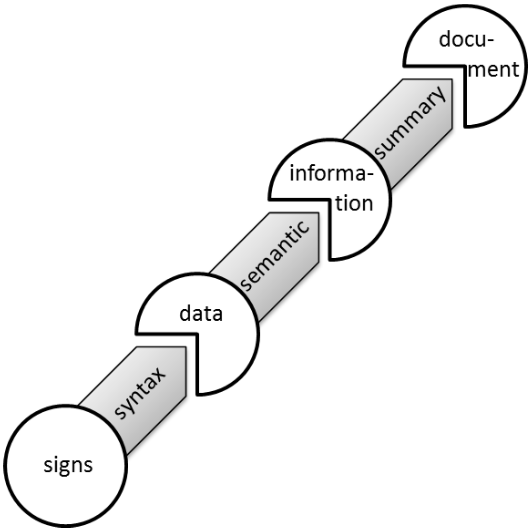

Figure 2: Difference of Documents and Data (based on [9, 10])

Data that come from several and independent IT tool is, by its own nature, heterogeneous, since each isolated application has the autonomy to generate and manage data, the integration of heterogeneous data requires conflict resolution of heterogeneity and the transformation of document and data sources into an integrated concept [11]. Data management systems for specific domains are quite matured, but not the collaboration among them. The more domains have to collaborate, the more problems in data integration arise, which have to be addressed. How data exchange itself is a difficult problem to solve, sharing information is an even bigger challenge.

Synchronous and asynchronous data sharing and data exchange using a federative environment shall be realized based on web services. The transformation of data formats, resolving technical and semantic differences and the integration and aggregation of data from multiple, isolated systems is one of the key problems. In particular, the semantic description gets in focus of interest. As described, data combined with semantic description is defined as information. As a conclusion, same data combined with a different semantic description results in different information. In this way, separating data from native documents is one challenge, the description of data in combination with appropriate semantics is a further one. To solve the described problems, the requirements as well as the concept of the FFDM will be described.

REQUIREMENTS FOR THE CONCEPT

FFDM has to get a tool supporting the methods of concurrent and simultaneous engineering in factory planning processes. The domains product development and production planning have to interact and the various domain-specific IT tools have to be integrated. Also the communication during the factory planning process has to be harmonized. Following the requirements of FFDM to achieve its goals will be described:

- FFDM does not replace established tools of domain-specific processes. FFDM has to integrate existing domain-specific tools and processes by managing the generated data (not documents).
- FFDM has to manage the structures of products, processes and resources and to link each of these structures to enable a factory overlapping description.
- FFDM has to be workflow controlled, fulfilling the methods of release, change and variant management to support collaborative engineering.
- Factory defining data has to be represented independently of domain-specific IT tools. Data have to be managed released from documents. In particular the description of structures released from documents is a big challenge.
- The consistence of data managed by FFDM must be ensured. During the factory planning process several loops of iterations will be passed; a continuous change of data is the consequence. To handle this, the consistence of data is fundamental.
- Concepts of roles and views have to be considered. Developers from various domains are users of FFDM and have to access to data in their context of role in the factory planning process. Also the usability has to be ensured by graphical user interface (GUI).
- FFDM must have an open and modular structure. The integration into the FFDM must be possible for suppliers from various domains.
- To realize the implementation of FFDM, a semantic information model must exist. This information model has do describe model and process specific information as well as the interdependences of these. This information model has to be understandable, modular and extensible.

CONCEPT OF FFDM

Core of the concept of FFDM is the domain-specific mapping of multi-dimensional structures and various views for analyzing of factory systems. FFDM has to be used as an integration environment for all distributed domain-specific IT tools, methods and processes. FFDM does not manage factory related documents; relevant data within the domain-specific documents are the point of interest. For representing data XML (eXtensible Markup Language) can be convenient. XML is an open, flexible and neutral data format which counts as one of the most propagated standardized language. The lifetime of data exceeds the lifetime of IT tools multiple times. Because of this and the described attributes, XML is qualified as a durable data format. To get the represented data in combination to semantic description, RDF technology can be used in combination with XML. The primary function of the data model of RDF is to describe semantic coherences of data. The widespread representation of RDF is XML.

In particular, a mapping of related data generated by various domain-specific tools builds an essential foundation factory planning processes. Mapping occurs on metadata level. Metadata are describing data and contain identifying and classifying data. The detected standard for exchanging metadata is Simple Object Access Protocol (SOAP). SOAP is an XML message protocol for service interoperability that ensures neutral metadata representation. Within these SOAP-messages, metadata as well as represented packaged (encapsulated) data (XML without RDF) or information (XML with RDF) attached to these SOAP-messages can be exchanged. Not only metadata, but also relevant queried data from documents can be exchanged due to SOAP messages. Because of the neutral information and data representation operating system independence is ensured.

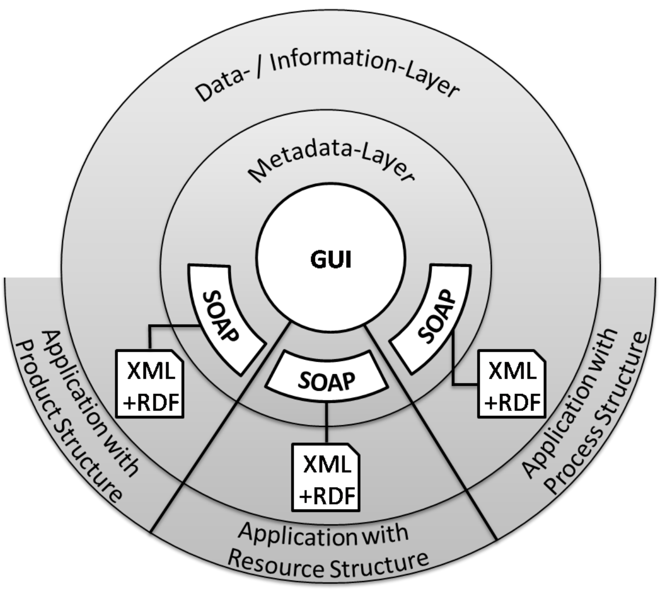

Figure 3: Concept of the FFDM

In general substantial characteristics of FFDM comrise:

- Factory structure and configuration,
- Release and change management,
- Production process structure and configuration,
- Identification and classification,
- Data retrieval and
- Collaborative engineering.

Factory structure is the core of an integrative concept of FFDM. All domain-specific data will be connected with elements of factory structure on metadata-level during the planning processes. Factory structure means in this context the integration of for product-, process- and resource-structure. To realize this integration of structures, a federative environment is essential. Existing data management systems for realizing integrated product-, process- and resource-planning often build upon a central database with monolithic data models [5]. Therefore such systems are inflexible in respect of the possibility of modifications and extensions for generating product-, process- and resource-documents and data [12]. As a consequence a necessity for a federative environment, which allows an open and neutral manipulation of product-, process- and resource-planning processes does exist.

The federative infrastructure is based on virtual integration of data and information from different autonomous source systems (e.g. CAD, PDM, ERP, PPC). The usage of the concept provides the local autonomy of the domains regarding the administration of tools and generated documents and data. A federative layer is in charge to join necessary information from various domain-specific source systems. For the user is no necessity of manipulating the architecture of the local system [13]. In addition to the representation in form of the federation layer, a standardized interface (Graphical User Interface - GUI) is necessary in order to access distributed data. This GUI has to be tool and system independent.

To get the federation environment implemented and work, a coherent information model for FFDM attached with appropriate functional web services e.g. for the exchange of product, process and resource data has to be specified. The federative environment is based on the exchange of metadata. In order to access metadata, web services are required, which allow queries and manipulation of data on different heterogeneous, domain-specific and local systems. These services provide a transmission of data between different data sources and mapping of semantic and syntax of source data to target (system specific) data.

ARCHITECTURE OF FFDM

The technical realization of a federative environment by using SOA is defined. The usage of this technology is a result of the requirements and the concept. The principle of SOA is based on loose coupling of services. The two main advantages are the independence of users of services from system specific applications as well as the flexibility of loosely coupled applications themselves. SOA is currently the most favored and promising approach. Within the IT environment, SOA can be described with following characteristics [14]:

- SOA is process-controlled and –driven. The integration of processes is the main target. The processes here are factory planning processes
- SOA is independent. No specific IT technology or environment is required for the implementation of SOA. Hereby SOA is extensible.
- SOA leads to an increasing of connectivity in heterogeneous environments (processes, data, and applications). The interconnection of various existing heterogeneous IT systems will be improved.

The requirements showed below must be attended due to these characteristics of SOA:

- integration of existing domain-specific systems and processes, not the replacing,
- managing and linking of process, resource and product structures,
- workflow controlled processes,
- independent representation of factory defining data and
- open and modular structure to be ensured.

To fulfill the transfer of metadata between the described heterogeneous systems, SOAP as a standardized transport protocol is dedicated in this concept (section "CONCEPT OF FFDM").

The architecture of FFDM builds upon four layers:
- Front-end layer (presentation layer),
- Federation layer (communication layer),
- Interface layer (web services) and
- Back-end layer (systems).

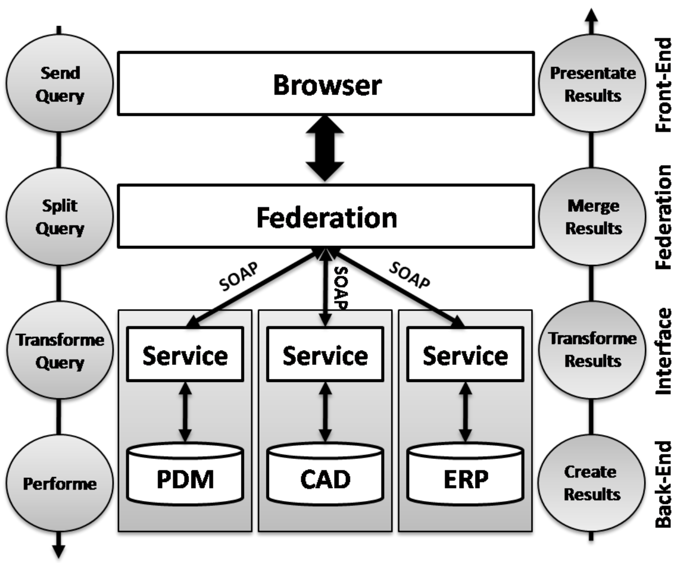

Figure 4: Architecture of the FFDM

The front-end accepts the request (input) of a user and presents merged response data of back-end systems. This kind of realization meets the requirements of system independence and a GUI, where roles and views could be implemented. Due to federative characteristics, the client sends queries to a central node passing the front-end layer. This central node is defined as a federation layer. The federation layer communicates with web services. These services are within the interface layer and are coupled with systems like CAD, PDM, ERP or PPC systems (back-end layer). The federation layer and interface layer can be regarded as a communication layer between front-end and back-end layer. The federation layer is a central node of virtual integration of services. The client does not need to send queries directly to several services individually. Queries will be divided into the service-relevant tasks and sent to the appropriate services. The services send the queries to the systems in a compatible (preferred neutral) form whereon native operation invocation of the respective system starts. The solved results by systems are documents and data. Only the generated data within the documents are relevant for exchange. After solving the queries by systems, the results will be sent to the federation layer by the service layer. The service layer acts as a wrapper, which envelopes the generated data and connects this encapsulated data as a SOAP message to the federation layer. Within SOAP-messages metadata can be exchanged and factory defining data can be attached as a XML-document. The encapsulated results of all queried services will be sent summarized to the client.

INFORMATION MODEL

Prerequisite for implementing the FFDM is an information model. The factory information model consists of a core model and various domain-specific partial models. The general structure of the FFDM is defined within the core model, which gives a description of all objects and their relations in an uniform format. The basic constructs of this model have to be described in the modeling language UML, the description of the dynamic part of the objects have to be expressed in an Object Constraint Language (OCL). Due to the dynamic part of the model, the object quality, quality changes and the course of events have to be specified. By using OCL a documentation of the model as well as a specification and formalization of the software functionality can be implemented.

IMPLEMENTATION

The concept of the implementation builds upon four levels, analog to the four layers of the architecture of the FFDM. Starting from the front-end layer, a browser has to be created. The browser is connected to web services. The connection is implemented with SOAP technology. The SOAP-messages transfer metadata about processes, products and resources. Data resulting of domain-specific documents are attached as an XML-document, added with semantic descriptions in RDF. The metadata and data of processes, products and resources have to result from documents of domain-specific applications.

FRONT-END

The task of the browser is to link and to structure process, product and resource data and metadata. The approach of the visualization of links and relations generates a matrix. An example for the visualization of coherences is the visualization of the relations between products with resources in a browser (figure 5).

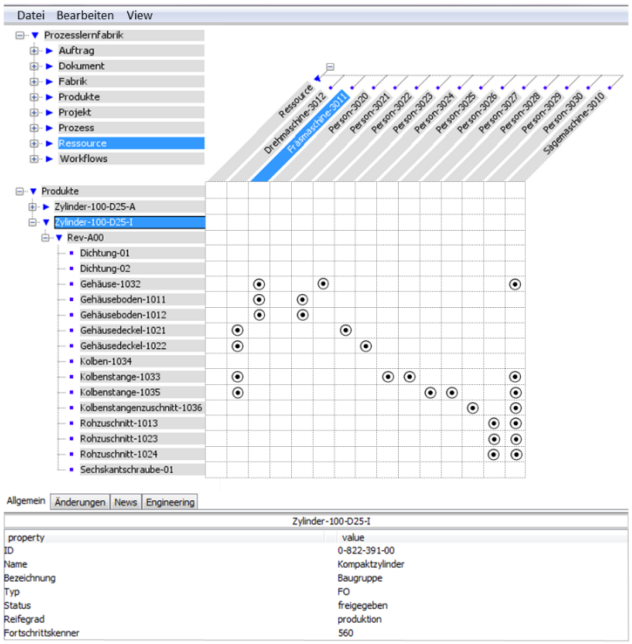

Figure 5: Example for a browser (front-end)

To explain following examples, exemplary data is assumed:

- *process milling and its variants 123 and 456,*
- *milling machine UVW,*
- *milling machine XYZ,*
- *person A and*
- *product D and its variants.*

Process metadata are information like milling, drilling etc., the attached data are machining time, set-up time etc. Resources are defined machines like *milling machine XYZ* or the worker *person A*. For the same *process milling*, the same resource *person A* can work at the different resources *milling machine UVW* and *milling machine XYZ*. Further interesting views are the relations between resources and products or processes and products. At the matrix products–resources the relations of produced products and the resources like machines or workers are visualized (compare to figure 5). Combined data like the machining time of a defined *product D* at the resource *milling machine XYZ* and the resource *person A* can be queried.

To fulfill the possibilities of variant management, various variants of factory structures should be visualized in a linked form of process, product and resource data. There are different variants within the described data. Processes for example have specific numbers of variants, products or resources itself, too. Within the browser relations between these variant and the data itself are shown. Example: *Product D* can be produced in *variant 456* of *process milling*, but not in *variant 123*. Within *variant 123* of process milling the defined resource *milling machine XYZ* is not able to produce this product. Within *variant 456* of *process milling* the resource *milling machine UVW* is defined, so *product D* can be produced. On the other hand several variants of *product D* can exist. The ability of producing these several variants with defined variants of processes and resources can to be cleared with the matrix browser.

METADATA IN SOAP

SOAP is a protocol for exchanging messages based on XML. The requirement of a SOAP-message representation is an envelope. Within this envelope, a body-element is required; a header-element is optional. Within the body-element, metadata from various domain-specific applications can be included. Included metadata are author, date, kind of application etc.

In the first phase of implementation, SOAP-messages are created by hand. The most common metadata are included in a representative form. The content and the structure of a representative SOAP-message result from research. The focus of the first phase is the functionality of the connection of front-end and SOAP-messages.

In further phases of this project, documents will be created as in practice is common. Documents will be linked with services, which describe metadata in an auto-generated form.

DATA IN XML + RDF

Analog to the description of metadata in SOAP-messages, data resulting of documents are described manually in a XML-document. The data described in XML are representative data, as implemented in domain-specific applications like CAD, PDM, ERP or PPC. These implemented data are described in applications as well as in XML-documents. In the first phase of the project, the description of data is redundant (application and XML), the focus is the visualization of data from XML within the browser. To get data combined with semantic description, the XML-document is extended with RDF content.

Getting data described in XML with RDF content generated from documents automatically and to visualize these information within the browser is the core of further phases of the project.

APPLICATIONS, DOCUMENTS AND DATA

As described in previous sections, the implementation of the concept is divided into two phases. In the first phase the focus is the implementation of the general process chain BROWSER – QUERY – SERVICE DATA/METADATA and return. Applications are not involved directly. The results generated by applications and described as data within documents is simulated by a description of data and metadata manually within SOAP, XML and RDF. To close of the gap APPLICATION – DATA/METADATA is not in core-interest of the first phase. This topic will be treated within the second phase of the concept.

GOAL

The goal of the concept of the FFDM based on SOA and semantic description with RDF is to meet the challenges of factory planning processes. One of the key-challenges is the integration of isolated, domain-specific IT tools and generated domain-specific partial models as well as the coordination and synchronization of engineering workflows. This integration is based on a document independent factory structure description linked with factory defining metadata. The goal of the first phase is to realize a following use case:

A user formulates queries by using the browser. These queries pass the federation layer and will be sent to the respective web services. The respective services parse the existing SOAP-messages as well as the attached XML+RDF documents. The parsed data will be sent back to the federation layer and afterwards visualized within the browser.

The goal for further phases is to extend the linking of applications. A user gives queries into the browser; these pass the federation layer and go to the respective services. The respective services transform the neutral queries into application-compliant queries. Applications execute these queries; the results are represented in data within documents. These data and metadata will be exported into SOAP-messages (metadata) attached with XML-RDF documents (data). The SOAP-messages will be collected within the federation layer and afterwards summarized and visualized within the browser. The browser should have the possibility to visualize the results in different views like product-, process- and resource-view.

In order to couple the different views, the suggested semantically coherent information model has to exist. The modeling of the information model occurs parallel to the implementation of the concept.

CONCLUSION

The described current situation and problems in factory planning processes give a rough overview of the challenges of enterprises. Factories can be considered as independent complex systems. Passing the phases of factory life cycle several participating domains have to be integrated [15]. This integration of all participating domains and their systems or IT-tools is the main challenge. The current situation is a heterogeneous IT-environment as a result of a disharmonious domain-specific factory planning process. Current isolated IT-tools are not prepared for a cross domain factory planning process. To face up to these deficits, the concept of FFDM is presented in this paper. Core of the concept of FFDM is managing and linking of domain-specific data (not documents or applications) based on SOA technology. Within the scope of SOA used in this context, SOAP-messages, XML- and RDF-technologies are in focus of interest.

The requirements for FFDM, which have to be fulfilled by this concept, have been described. In general the target of the requirements are the development of a tool supporting the methods of concurrent and simultaneous engineering, the integration of the domains product development and production planning and the support of the communication during the factory planning process.

Research on the information model as well as the implementation of the concept has started already and the progress as well as the outlook of the further implementation concept is described.

The advantage of the concept is to use existing technologies (web services, XML, RDF, SOAP) as well as the integration of the concept into an existing IT environment. Existing domain-specific IT-tools and processes can keep their autonomy combined with the integration of the generated data and the exchange of metadata in a central integration platform. The result of using existing standards and keeping of the domain-specific environment is the improvement and harmonization of factory planning processes considering the costs for investment for industrial companies.

ACKNOWLEDGMENTS

Acknowledgments to DFG (Deutsche Forschungsgemeinschaft, Germany) and CAPES (Coordenação de Aperfeiçoamento de Pessoal de Nível Superior, Brazil) for funding this project within the BRAGECRIM (Brazilian - German Collaborative Research Initiative on Manufacturing Technology) Initiative.

REFERENCES

[1] Langenberg, L.: Firmenspezifische Wissensportale für die Produktentwicklung, Book, Shaker Verlag, Aachen, Germany, 2001.

[2] Jania, T.: Änderungsmanagement auf Basis eines integrierten Prozess- und Produktdatenmodells mit dem Ziel einer durchgängigen Komplexitätsbewertung, Dissertation of the University of Paderborn, Paderborn, Germany, 2004.

[3] Aggteleky, B.: Fabrikplanung – Werksentwicklung und Betriebsrationalisierung, Book, Carl Hanser Verlag, München, Germany, 1987

[4] Kettner, H.; Schmitt, J.; Greim, H. R.: Leitfaden der systematischen Fabrikplanung, Book, Fachverlag Leipzig, Leipzig, Germany, 1984.

[5] Anderl, R.; Rezaei, M.: Federative Factory Data Management – An approach based upon Service Oriented Architecture (SOA), Book, International CIRP Conference - Digital Enterprise Technology, Setúbal, Portugal, 2006.

[6] Anderl, R.; Rezaei, M.; Schützer, K.; Moura, A.: Journal, ProduktdatenJournal (2), pp. 49-53, Germany, 2009.

[7] Frankel, A.: Advanced Tooling Solution, Paper, 10. Seminário Internacional de Alta Tecnologia, SCPM, UNIMEP, Piraciba, Brazil, 2005.

[8] Zenner, C.: Durchgängiges Variantenmanagement der technischen Produktionsplanung, Dissertation oft he University of Saarland, Saarbrücken, Germany, 2006.

[9] Deutsches Institut für Normung (DIN): DIN 6789, Teil 1, Dokumentensystematik – Aufbau Technischer Produktdokumentationen, Beuth Verlag, Berlin, Germany, 1990.

[10] Probst, G.; Raub, S.; Romhardt, K.: Wissen managen: Wie Unternehmen ihre wertvollste Ressource optimal nutzen, Book, Gabler-Verlag, Wiesbaden, Germany, 2003.

[11] Lubell J., Peak R. S., Srinivasan V., Waterbury S.: STEP, XML and UML: Complementary Technologies, Paper, ASME 2004 Design Engineering Technical Conferences and Computers and Information in Engineering Conference, DETC2004-57743, Salt Lake City, USA, 2004.

[12] Fröhlich, A.: Integration des Prüffeldes in die virtuelle Produktentstehung, Dissertation of Technische Universität Darmstadt, Shaker Verlag, Aachen, Germany, 2003.

[13] Shet, A.-P.; Larson, J.-A.: Federative Database Systems for Managing Distributed, Heterogeneous and Autonomeous Databases, Book, ACM Computing Surveys, Vol. 22, No. 3, 1990.

[14] Abramovici, M.; Bellalouna, F.: SOA-Architecture for the Integration of Domain-Specific PLM Systems within the Mechatronic Product Development, Paper, Proceedings of the 7th International Symposium on Tools and Methods of Competitive Engineering, TMCE 2008, Izmir, Turkey, 2008.

[15] Westkämper, E.; Constantinescu, C.; Hummel, V.: New paradigms in Manufacturing Engineering: Factory Life Cycle, Annals of the Academic Society for Production Engineering, Research and Development, Vol. 13, No. 1, 2006.