

Categorisation of Data Management Solutions for Heterogeneous Data in Collaborative Virtual Engineering

Stephan Vornholt
Faculty of Computer Science
University of Magdeburg
Magdeburg, Germany
vornholt@iti.cs.uni-
magdeburg.de

Ingolf Geist
Faculty of Computer Science
University of Magdeburg
Magdeburg, Germany
geist@iti.cs.uni-
magdeburg.de

Yuexiao Li
Faculty of Computer Science
University of Magdeburg
Magdeburg, Germany
yuexiao.li@st.ovgu.de

ABSTRACT

Computer based virtual development is the basis of product engineering, called virtual engineering. The increasing demand of virtual engineering solutions for new domains and companies leads to a huge width of approaches in this area. To enhance the communication and collaboration between engineers, engineering processes and tasks are supported by specialized applications. Each of the collaboration solutions has benefits and weaknesses, depending on the tasks. This paper analyzes collaboration approaches, i.e., different exchange and integration techniques for the virtual engineering domain. To enable the classification and comparison of existing data management solutions, collaboration techniques and technical approaches to handle heterogeneous data are categorized. The developed categories will help choosing or developing new collaboration systems.

Categories and Subject Descriptors

K.4.3 [Organizational Impacts]: Computer-supported collaborative work, collaborative computing; H.2.5 [Heterogeneous Databases]: Data translation; J.6 [Computer-aided engineering]: Computer-aided design (CAD), computer aided manufacturing (CAM)

Keywords

Virtual Engineering, Data Management, Integration, Exchange, Collaboration, PLM, PDM

1. INTRODUCTION

Several computer based applications (e.g., computer aided quality, -design, -product planning) form the basis of product development (see Figure 1). Together with virtual reality and simulation, they influence design decisions in the whole Product Life Cycle (PLC). The Virtual Engineering (VE) process is part of the PLC containing all phases

of product development from concept, design, and analysis/simulation, up to manufacturing. Cooperation of different fields allows early production planning, design process management, and pre-defined personal management. The collaboration of engineers from different domains (e.g., mechanical, electric, thermal) promises a faster and cheaper development of higher quality products.

Two main approaches were realized to improve the cooperation in VE processes:

Data Management: Solutions for Product Data Management (PDM) and Product Lifecycle Management (PLM) have been developed. PDM and PLM systems manage the meta-data and data for the complete life cycle of a product. Once developed for the process and role management, they were extended into an application integration platform (described in [10], [2]).

Application Integration: Applications encapsulate more domains. On the one side domain specific applications were enhanced by adding new scopes, on the other side different solutions are merged into one application. Several design applications for example integrate simulation engines as well as their own document management system (e.g., Pro/Engineer or SolidWorks).

Every specialist still requires his own (often traditional) view on the data and the geometry model for product representation. Because of different representation forms, data formats are heterogeneous even for applications of the same domain. Hence, product definitions in one domain are often incompatible to other domains. However, information has to be exchanged, which leads to input failures and cost intensive remodelling of product information. Therefore, solutions for the (semi-)automatic exchange of data have been developed. To overcome exchange weaknesses, other solutions like combined applications, data based exchange of information, or the integration into a common data format have been suggested.

Those exchange and integration solutions have an enormous benefit for the cooperative product design. However, the required data handling solution depends on the task and collaboration form. Some companies can only afford a small bundle of applications or do not have the time to implement a complex system [11]. The period of vocational adjustment is also a cost factor, when changes and updates of applications and technologies are considered. Applications and their combination have to be adapted to a set of tasks. The problem is to find the right solution for specific requirements.

Permission to make digital or hard copies of all or part of this work for personal or classroom use is granted without fee provided that copies are not made or distributed for profit or commercial advantage and that copies bear this notice and the full citation on the first page. To copy otherwise, to republish, to post on servers or to redistribute to lists, requires prior specific permission and/or a fee.

IWDE 2010 Magdeburg, Germany

Copyright 2010 ACM 978-1-60558-992-3/10/06 ...\$10.00.

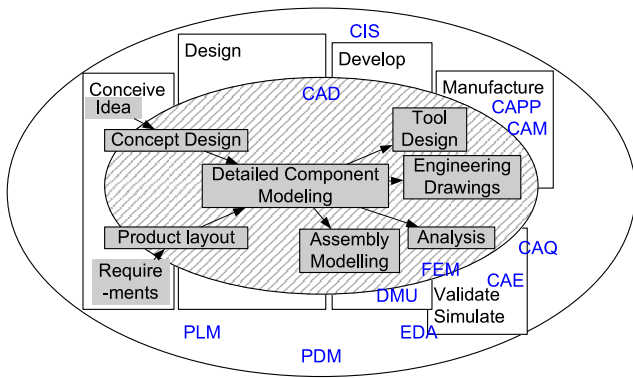


Figure 1: Various Computer Aided Technologies

Up to our best knowledge there does not exist a detailed categorisation for collaborative engineering techniques considering the technical implementation. Existing surveys focus on management systems or present a small sector to localise own survey scopes. To bridge this gap, we give an overview of different requirements, approaches, and solutions. We propose categories of existing works in the field of data exchange and integration in Virtual Engineering. Kinds of collaboration and modelling of VE domains are presented in Section 2, starting with the fundamentals of Virtual Engineering. Due to increasing amount of needs and solutions we identify requirements for the support of collaboration. In Section 3 we present the two basic methods data exchange and integration as well as shared approaches. After identifying kinds of collaborative data management in Section 4, we classify and compare exemplary solutions. Finally, a conclusion and perspectives are presented in Section 5.

2. PRELIMINARIES

2.1 Virtual Engineering

On the basis of virtualization (i.e., geometric modeling) Jian describes Virtual Engineering in [9] as user-centred process that provides a collaborative framework to integrate all design models, simulation results, test data, and other decision-support tools in a readily accessible environment.

Bullinger [3] identifies five levels of the VE process:

1. **Data creation:** Geometry data is created as foundation of continuous operations in all following processes. All CAx and EDM / PDM data is created and stored in files, e.g., CAD-file, FEM-file.
2. **Data management:** EDM/PDM systems store, transfer, and administrate information about used applications and processes.
3. **System integration:** includes data integration as well as the connection to applications. All applications, which are required by users, are integrated. Data exchange should be handled bidirectional, fast, and easily, e.g., between CAD and VR-Simulation.
4. **VE organization:** This level comprises integrated process management. It includes communication and

management of finite (qualitative and quantitative) resources, handle project and process risks and difficulties as well as tracking project advancement.

5. **Application access:** The level comprises communication between costumers, cooperating companies, and external engineering partners. However, integration of new domains and management of complexity is also affected by status control, portals, and access control.

2.1.1 Product Lifecycle & VE Process

Virtual engineering is influenced by idea generation, product development, production, marketing, up to waste management. However, most VE applications focus on the data creation process that is illustrated as follows: The VE process is based on a conceptual model which includes a graphical representation and dependencies of subparts as well as constraints and domain descriptions. The conceptual model is converted into a CAD model. Different development domains (electrical, mechatronical, thermal, ...) are successively added. The connection between different subparts is defined in each domain. The resulting virtual prototype can be converted into different simulation based formats (FEM simulation, mechatronic simulation, fluid dynamics, ...). In each of these steps, standards and libraries with existing products are used to increase correctness. Changes on the intended product, like variants or versions require a re-design of the product. In summary, the product development becomes a non-linear process.

Simulation results identify weak spots or deficiencies and give information for further improvements. Typically the following analysis domains comprises the backbone of product analysis:

Mechanical simulation models are used for analyzing the mechanical behaviour of prototypes, e.g., kinematic behaviour, testing of possible collisions, and allowed movements.

Mechatronic analysis [6]: the mechatronic model adds electric drive and control components to the mechanical model.

Finite element (FE) analysis models are used to analyze the elastic behaviour of certain system parts. Typical tasks of FE analysis are the computation of resonant frequency or bending stress.

Economic management, i.e., marketing, production planning, cost, logistics, and inventory. Here, products constraints can be adapted and economic, logistic or technical plans created.

As result of the process instantiated models, called **Virtual Products (VP)**, can be derived, shown, and tested with different simulations.

Virtual Prototyping focuses on the presentation, interaction, and graphical evaluation of models. Virtual reality techniques are used to enable training, preview on products, or other graphical based representations for costumers.

2.2 Collaboration

Cooperative Engineering is a systematic approach to enhance the integrated, concurrent design of products and their related processes, including manufacturing and support [14]. Following Prasad [15] cooperative team work includes seven elements: communication, compromise, commitment, continuous improvement, consensus, coordination,

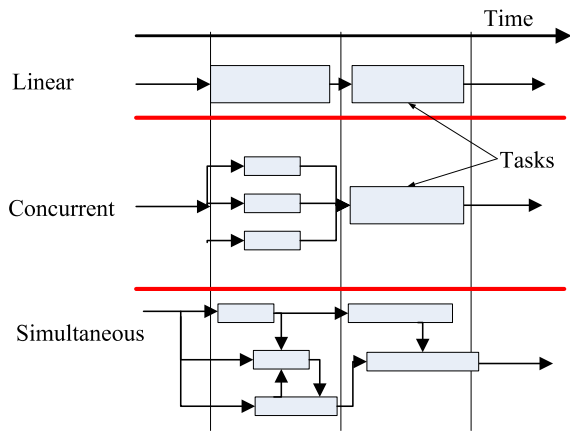


Figure 2: Kinds of Collaborative Engineering

and collaboration. These elements describe the level of interaction in the VE process. We will focus on collaboration, which Prasad defines as follows: Participants work on shared objects to seek out the unplanned and unpredictable.

The basic types of collaboration in engineering can be identified as (cf. Figure 2):

Linear engineering: The complete developing process is executed step by step according to the order of time. The procedures in each phase are developed consecutively.

Concurrent engineering: The work in one phase is divided into different parallel working domains that are independent from each other (e.g., electrical model and mechanic model). The resulting models might require a final composition (e.g., mechatronic model). Concurrent engineering is a special form of simultaneous engineering.

Simultaneous engineering: In this approach parallel processes depend on each other and exchange information when needed. A dependent work process can also be stopped when necessary information is not available.

2.3 Data Management & Models

2.3.1 Example: Light bulb

The model of a light bulb, illustrated in Figure 3, is investigated as an exemplary virtual product designed using VE. The intuitive picture comprises electrical, geometric, thermal, and physical components which could be described in detail in different domains to simulate the behaviour of the light bulb. An electrical model focuses on electrical wires and contacts, while the thermal model focuses on gas and pressure. However, the dependencies (inductor/coil and pressure/temperature) require cooperative design.

2.3.2 Modelling

Feature-based modelling : In today's common understanding, features are the main modelling objects of CAD systems. Regarding their semantics and shape content, features in modern CAD systems can be considered as being complex design objects, characterized by a set of attributes or properties that determine the resulting behaviour of the

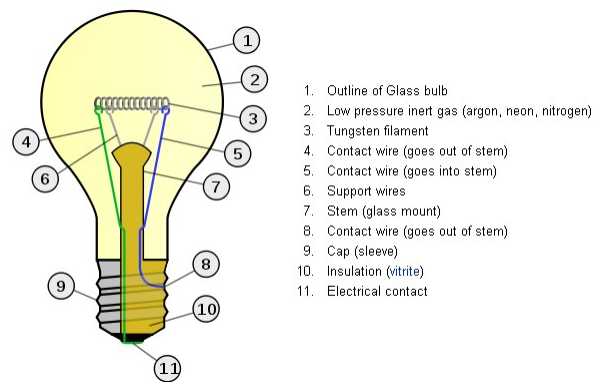


Figure 3: Lightbulb Structure Model [8]

feature within the CAD model and, specifically, the shape model. The semantic content varies with the scope of domain and application. Features for design, assembly, manufacturing, or quality assurance contain different application specific information. [17]

Parameter-based modelling : CAD systems provide parametric design both in shape and feature modelling. The user may assign values to dimensional and feature variables. From the user point of view, geometric constraints, e.g., shape elements being parallel, intersecting, co-planar, rectangular have to be distinguished. Also constraints on parameters owned by features, part or assembly objects need their own description. Additionally, CAD systems provide means for defining shape or feature independent variables.[17]

2.3.3 Problems during data exchange

Systems for collaborative engineering also have to consider common failures. As shown in [4] and [18] the main problems occurring within data exchange are:

Lost data: data loss during data transformation occurs when information is mapped from the source to the target tool, with no equal or suitable structure. This problem is hard to solve when different data structures are mapped.

Incorrect interpretation: Cognitive and naming heterogeneity as well as similar names often result in misinterpretations. A more understandable problem is a different granularity of decimal places or different semantic units.

Integration of new/changed domains: When new or updated applications are integrated to design a product, common environments require an adaptation. The adaptation affects existing structures as well as transfer protocols.

2.3.4 Requirements

There is a lack of methods for data exchange between various tools in VE. Therefore, integration of design, analysis, and simulation tools becomes important as a research topic. Thus a fault-tolerant system should ensure **trust-worthy data** and any possible errors should be quantified. Since a VP describes the product data from different aspects, it should provide a **collaborative** design environment. An open architecture is required, where different **new domains**

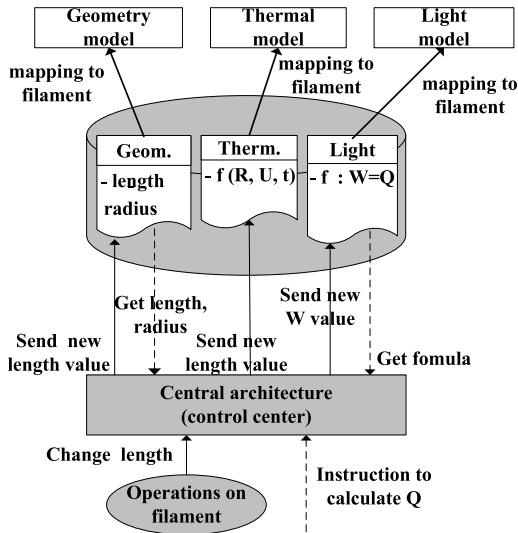


Figure 4: Control Center

from VE can be integrated. Then it helps to get the necessary, combined information.

3. KINDS OF DATA MANAGEMENT

To provide an overview of existing approaches, the following section presents ideas of systems supporting collaborative engineering. Oh et al. [13] describe basic approaches for data integration in a sorted way: manually re-input, file exchange, stand alone databases and shared databases or repositories. Roucoules [16] describes on the basis of a cooperative design environment the integration levels: interface, application, engine and database. In the following we present, a fine granular categorisation. We describe based on general specification for every collaboration system (propagation time, data, and exchange control), typical with exchange or integration solutions coupled approaches.

General specifications in every system (integrative or exchange) are:

Propagation data (Automatic vs. On Demand): Some solutions identify and propagate changes instead of translating the complete model. These systems use prepared dependencies to identify and compare parameters of different data formats. The benefit of these systems is the concentration of all involved engineers on new parts instead of a new version even for uninvolved domains. Change identification is mainly done in system specific data exchange solutions or managed by a control center. Figure 4 demonstrates the function of a control center for dependencies between geometry, thermal, and light model of the light bulb example. Continuous arrows depict the parameter change of filament (length) operations. The control center detects the dependent models and sets (sends) the new length, for the light model, the parameter W is changed based on a formula function in the control center.

Propagation time (Changes vs. Complete): Solutions vary in the time of propagating changes. Some use an application or middleware to control and propagate changes

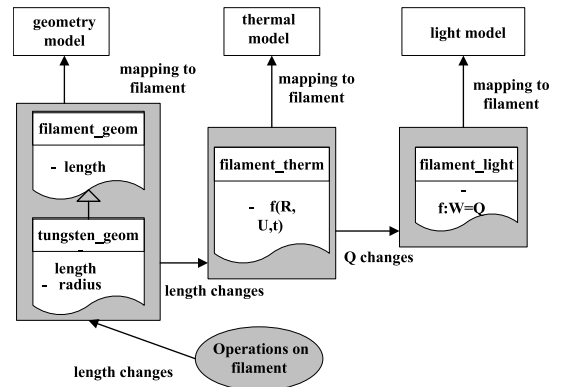


Figure 5: Geometry Change

each time the data is stored or changes were detected, while other systems propagate them on demand. If a process management is used and access roles are defined, a controlled automatic notification and data transfer to affected designers is possible. Architectures to control dependencies or role management to identify affected engineers are necessary to ensure only affected and not every involved data will be exchanged. Within concurrent or simulations engineering, automatic uncontrolled data transfer for frequent parameter changes constrains work.

Exchange control (Consecutive vs. Central): Consecutive transactions use neutral or system specific exchange formats to connect applications by forwarding changes in a sequence of translations. Systems forward changes to affected domains. Therefore less translation functions are required. Figure 5 demonstrates consecutive transaction with the light bulb example. A change of the parameter filament (length) is propagated first. The change is integrated into the geometry model, which translates the changes value into the thermal model. The light model, based on the thermal model data gets the new parameter from a known data format. The opposite is **Central Transaction** management, e.g., by an control center as presented in Figure 4.

3.1 Exchange Solutions

Data exchange is defined as the process of transferring relevant/common information between different engineering parties in order to meet the projects objectives and to minimise data re-entry and duplication [1]. Data transfer requires exchangeable data mediums, bus-systems, direct connections, networks, or remote transmission. It depends on at least one source and target system. However, functions to transfer data into the target component or application are required. According to the format the systems take to transfer the data, different kinds of data exchange are classified as following:

- **System specific** data exchange (Figure 6(a)) includes the data transfer between two applications.
 - Direct data **transfer** from one system to another and the reverse process: The programs for each point-to-point transfer are written specific to each transfer. So an external transfer program is necessary.

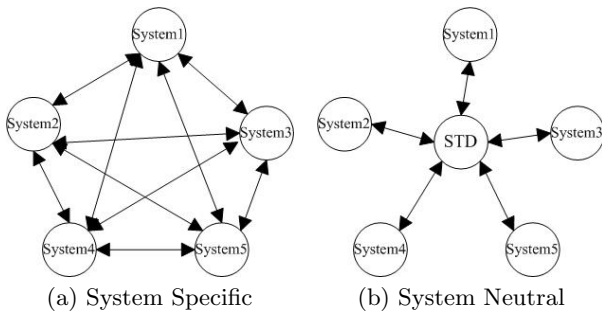


Figure 6: Kinds of Exchange

- Systems have specific **functions** or subcomponents to export data into external formats. This situation is limited to specific systems and not generally supported. Especially, it is recommended for transfer of data designed using a common architecture.
- **System neutral** data exchange (see Figure 6(b)) translates all models into one format. This common format is named as neutral format. Standards like IGES and STEP have been developed.

3.2 Integration Solutions

Integration of information or data includes merging information of heterogeneous data storage systems in one common data structure. In difference to exchange solutions, integration solutions share data. The applications still use a local format, because the common data structure might be less detailed or concentrates on common information. Figure 7 shows one integration example. The applications for geometry and thermal model have their own database where information is stored using XML (geometry) or in objects (thermal model). A common database integrates both information and the dependencies between the models. In the following we describe approaches for integrated engineering.

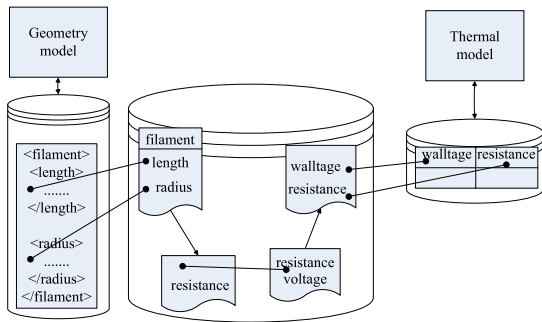


Figure 7: Integration Example

3.2.1 User Interface

Obviously each domain expert requires a representation of his data suitable for his domain. In consequence, **Separated Views** are defined. Each domain / application has its own data representation, independent from data format. Often integrated systems use a common interface to enable

global view on the data and combine data for new information. **Common Interface (Integrated Interfaces)** for applications (e.g., Figure 4) can represent data based on different domains. Interface integration is based either on PDM systems, which can recognise CAD files and launch them to the CAD system or make some PDM functions available via CAD menu. Or it is based on a new interface independent from the CAD and PDM system. For example, common user interfaces enable the extension of an original geometric model by adding an thermal model to get a more complete view on the model (see Figure 4).

3.2.2 Architectures

In the following, three architectures for data integration are presented:

Common Objects (Figure 8(a)): A common data model is used for different applications. A translator (half-link) for the systems translates the formats into a common data model and back to the local format. When we need or store new data, the common database is used. Similar to the exchange formats we translate and integrate changes or complete data.

Managed data (Figure 8(b)): This approach integrates models (e.g., mechanical analysis model) in one common structure (e.g., data warehouse), though still in different formats. The dependencies are controlled by an external application which manages common parameters and constraints. The management is independent from transfer or common data models. Only dependencies between heterogeneous models are managed.

Inclusion (Figure 8(c)): If one system integrates another system (e.g., a CAD system integrates simulations), then one data model is integrated (e.g., internal translation) in the main system. The applications user interface is integrated, too.

Based on the suggested data integration form, architecture, and intended support form, we decide how to store the data (**storage form: (database, data warehouse, or files)**).

3.2.3 Granularity of dependencies

The granularity of modelled dependencies is divided into three categories:

- **Data Management** (e.g., STEP, Component DB [18]): The data management focuses on the data itself. The parameter values or the different models are stored and managed. Dependencies can be defined on the basic level of variables. A common model structure or integration is necessary.
- **Document and Process Management** (e.g., PDM, PLM): The documents and files are managed. Dependencies of documents (e.g., version, variant) are managed as well as access and responsibility roles. The document management often integrates process management.
- **Meta-Data Management** (e.g., link DB [7], Ontologies): Here, meta-data, like the structure of the products, the contained models, and vocabulary, is used to define dependencies between different models.

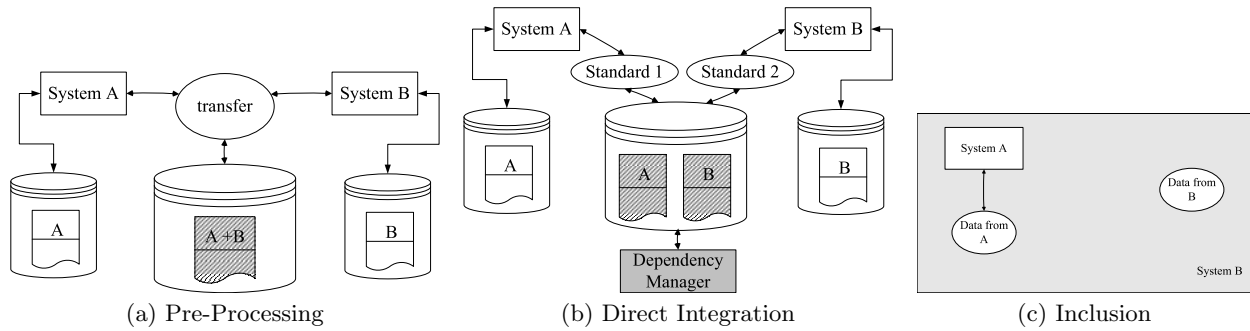


Figure 8: Kinds of Integration

3.2.4 Access

For integration it is necessary to obtain information from local applications. The access level for the integration is a characteristic for data exchange and integration methods. The different approaches are:

- **File:** Files of the systems are used to exchange information. For example: Figures 8(b) and 8(c) use an external transfer application based on files.
- **Database:** The exchange system uses the storage database or internal storage system of the applications, e.g., used in Figure 7.
- **Application:** The application itself is accessed and internal or external functions used to get the data 8(c).

3.3 Categories

We summarise the categories and present the main dependencies between them. Figure 9 illustrates the categories. Rectangles represent categories and ellipses specific approaches. The lines illustrate dependencies that have to be considered during decision making.

On the first level the collaboration type is defined (linear, concurrent or simultaneous). Based on this level either an exchange or an integration solution is selected (second level). While exchange is recommended as simple solution for linear collaboration, simultaneous engineering requires integration. Concurrent engineering uses a mixed form or chooses one based on expected communication intensity. Exchange solutions are defined by format, control, and propagation approaches. On the other side, integration solutions can be defined in all given categories except exchange control.

Some approaches require other categories: A central exchange control requires a central architecture (exchange control central → architecture). The storage form is based on the granularity level of dependencies. Files are arranged in document management systems while databases are used to store data or meta data. Data warehouses integrate data and process information. Each system can support different specifications of one category. An architecture including different systems can easy define a common interface (inclusion → common). Furthermore, common object architecture has to consider a data format.

Linear engineering (dark-gray) with data exchange is based on consecutive forwarding of system specific data. It

is completely forwarded to the next domain when the work of the first domain is finished (on demand).

Simultaneous engineering (light-gray) requires lots of data exchange and thus an integration solution is preferred. To support the engineers, an automatic exchange of changes enables current data for affected domains. Common objects stored in a database or data warehouse furthermore enable the storage of data and metadata as well as defined separated and common views on them.

4. CLASSIFICATION & DISCUSSION

The presented categories of Subsection 3.3 are used to classify three exemplary collaboration solutions (STEP, PLM, CCI) and results are in order to compare predefined requirements and systems.

4.1 Exemplary Classification

STEP is an ISO standard for product data exchange, storage and access. It has been developed based on previous standards (such as IGES), it extends and includes existing standards, but is independent from different implementations [5]. Based on the language, which redefines derivatives from CAx models, it extends them by the integration of application protocols, to integrate constraints and requirements for data models.

A bidirectional integration process using STEP is illustrated in Figure 10.

- 1) **Pre-processor:** The pre-processor is used to get the model from the CAx system.
- 2) **Integration:** Data model and application protocol is stored in step files.
- 3) **Translation (Postprocessor):** Translate the model from STEP format into the forward format, using a post processor.

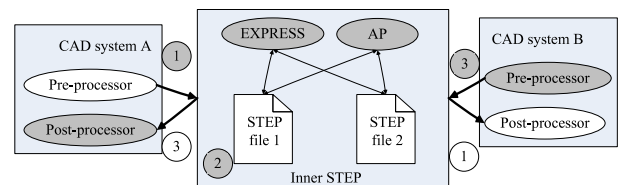


Figure 10: Data Exchange in STEP

The STEP database also illustrates that data exchange and data integration are not disjoint categories. Even when

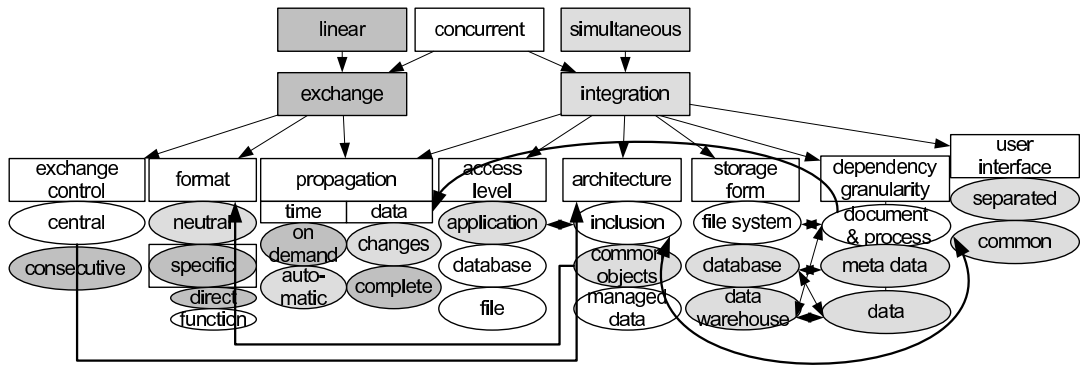


Figure 9: Categories for Collaborative VE

the concept behind both solutions is different, standards can be used as data model for the integration.

Figure 11 shows the classification of linear collaboration using STEP. Files are propagated on demand. STEP files exchange all existing data between the systems and the format is system neutral. Since designers use their own application, the interface is not shared. The designers work on files and exchange data without meta-information.

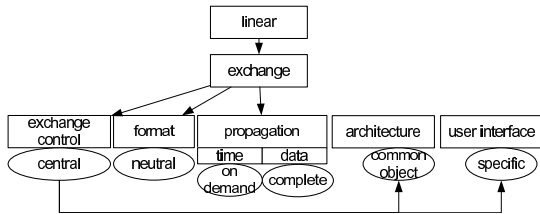


Figure 11: Classification STEP

PLM/PDM systems manage the product life cycle and optimize the interaction of product design, manufacturing, and life cycle activities. Product structure and composition is automatically derived from CAD systems.

PDM systems (e.g., Windchill designed by PTC) are used to control the product information and processes in the PLC. The functionalities in those systems are wide spread. They provide 3D visualisation and common user interfaces of CAD systems as well as the models. Some applications are integrated, e.g., for Windchill: MCAD-Tools, AutoCAD. Furthermore, it is connected to many mechanic and electronic CAD applications by external functions. The granularity is limited to data management approaches. However, the data item storage is less integrated and dependencies between parameter values require further external or internal defined functions.

Figure 12 describes a typical PDM/PLM solution. Managed exchange, concurrent, and simultaneous work is supported. Furthermore, it enables common as well as separated views. However it is limited to the document and meta management dependency level.

Integrated Database, e.g., Concurrent Component Interface (CCI): Data incompatibilities and inconsistencies between legacy and new database schemata exist [12], so in order to deal with these problems the extraction of data from legacy systems and integration into new systems

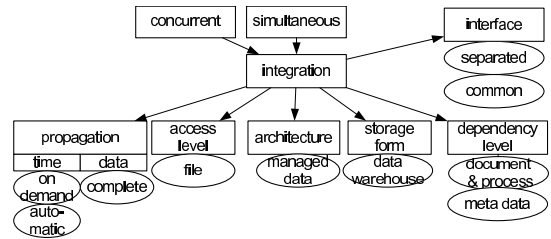


Figure 12: Classification PDM

are necessary. The incompatibilities may be semantic or quantitative. In [18] the authors describe a data integration model based on components which contain heterogeneous model parts. The containers for different part models have their own definable interface to define parameters. The main task of this approach is to enable consistency control between different models on the parameter level by considering the possibility of a separate view for every designer and easy definition of combined views.

Figure 13 classifies the described component interface integration solution. The scope of this approach lies on the data dependency level with common objects and propagated data changes. Thus, it can be used for concurrent and simultaneous engineering.

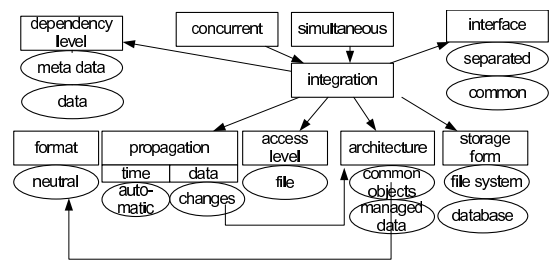


Figure 13: Classification CCI

4.2 Discussions

Based on the presented requirements in 2.3.4 typical advantages of the exemplary classified concurrent support applications are discussed:

All time **thrust-worth data** means to an engineer to work on the current data. He will be informed when changes occur. So the collaboration needs managed dependencies or a

	CCI	Exchange standard	PLM PDM
always trust worthy data	yes	no	limited
collaboration	simultaneous	linear	simultaneous
new domains	complex	2 HL or 2*n HL	complex
granularity	parameter	parameter	structure

Table 1: Comparaison

common data model with automatic propagation of changes. This characteristic is only existing at integrated data models and limited to the information of changes (if the engineer is affected by them or not) using PDM solutions. The **supported collaboration** form by exchange standards is limited to data transfer. CCI and PDM systems enable simultaneous work. However, simultaneous and concurrent solutions should use a form of integration and a common storage system. **Integration of new domains** into PDM and CCI are complex because a new common structure have to be defined. System neutral exchange standards are easy to maintain, while system specific standards need two half links per connected system. While the **granularity** of exchange standards and CCI enable the access on the parameter, PDM systems are limited to the structure and transfer of files.

5. CONCLUSIONS & PERSPECTIVES

This paper presented a categorisation of approaches for concurrent virtual engineering. Based on three types of collaboration, the main approaches integration and exchange of product data were presented. Those categories and their dependencies as well as different possibilities to handle them were illustrated. This categorisation enables, as shown in the last section, an overview and classification of different applications. On this base, it is possible to identify advantages and disadvantages of approaches in collaborative virtual engineering.

Previous considerations show that many possibilities have not been elaborated and new approaches for adaptable data management methods will still be needed. The described categories enable the classification of existing collaborative virtual engineering systems as well as the identification of weak spots where adapted solutions do not exist. This is a first step for a guideline to find the optimal solution for each possible engineering cooperation type.

6. ACKNOWLEDGMENTS

This work is supported by European Commission: EFRE COMO C1-3201201 und C3-320120.

7. REFERENCES

- [1] M. Alshawi and B. Ingirige. Web-enabled Project Management: an Emerging Paradigm in Construction. *Automation in Construction*, v12 i4, pages 349–364, 2003.
- [2] R. Anderl and J. Malzacher. SimPDM - Simulationsdatenmanagement-Standard nach Mass. *CAD CAM*, 2009(1-2):38–41, 2009.
- [3] H.-J. Bullinger. Virtual Engineering - Neue Wege zu einer schnellen Produktentwicklung. *Sonderforschungsbereich Entwicklung und Erprobung Innovativer Produkte - Rapid Prototyping (Forschungsforum)*, 2002.
- [4] R. Eckert, W. Mansel, and G. Specht. Model Transfer Among CASE Tools in Systems Engineering. *System Engineering*, 8(1):41–50, 2005.
- [5] J. Fowler. STEP for Data Management, Exchange and Sharing, 1995.
- [6] P. Fritzson and V. Engelson. Modelica - A Unified Object-Oriented Language for System Modelling and Simulation. In *ECCOP 1998*, pages 67–90, London, UK, 1998. Springer-Verlag.
- [7] I. Geist and S. Vornholt. Eine Link-Datenbank zur Integration von Virtual Engineering-Daten. In *Grundlagen von Datenbanken*, pages 45–49, 2009.
- [8] General Electric Company, Nela Park, Cleveland, OH. *Incandescent Lamps*, 1964. Publication Number TP-110.
- [9] C. Jian, D. McCorkle, M. Lorra, and K. Bryden. Applications of Virtual Engineering in Combusten Equipment Development and Engineering. *Research in Engineering Design*, 2006.
- [10] H. Kropsu-Vehkaperä, H. Haapasalo, J. Harkonen, and R. Silvola. Product data management practices in high-tech companies. *Industrial Management and Data Systems*, pages 758–774, 2009.
- [11] L. Laemmer and R. Bugow. PLM Services in Practice. In *The Future of Product Development*. Springer Berlin Heidelberg, 2007.
- [12] A. M.R.Abbasifard, M.Rahgozar and P.Pournemati. Using Automated Database Reverse Engineering for Database Integration. *World Academy of Science, Engineering and Technology*, 19, 2006.
- [13] Y. Oh, S. Han, and H. Suh. Mapping Product Structures between CAD and PDM Systems using UML. *Computer-Aided Design*, 33:521–529, 2001.
- [14] J. Pennell, R. Winner, H. Bertrand, and M. Slusarczuk. Concurrent engineering: an overview for Autotestcon. *AUTOTESTCON 89. IEEE Automatic Testing Conference. The Systems Readiness Technology Conference. Automatic Testing in the Next Decade and the 21st Century. Conference Record*, pages 88–99, 1989.
- [15] B. Prasad, F. Wang, and J. Deng. Towards a Computer-Supported Cooperative Environment for Concurrent Engineering. *Concurrent Engineering*, 5(3):233–252, 1997.
- [16] L. Roucoules and S. Tichkiewitch. CoDE: A Cooperative Design Environment- a New Generation of CAD Systems. *ConcurrEngineering Res Appl*, 8(4), 2000.
- [17] G. Spur and F.-L. Krause. CAD-Technik: Lehr- und Arbeitsbuch fuer die Rechnerunterstuetzung in Konstruktion und Arbeitsplanung. 1984.
- [18] S. Vornholt and I. Geist. Flexible Integration Model for Virtual Prototype Families. In *Proceedings on the 5th International Conference on Product Lifecycle Management (PLM08)*. Inderscience, July 2008.