



HAL
open science

Towards Integration of CAx Systems and a Multiple-View Product Model ler in Mechanical Design

Huijun Song, Benoit Eynard, Pascal Lafon, Lionel Roucoules

► **To cite this version:**

Huijun Song, Benoit Eynard, Pascal Lafon, Lionel Roucoules. Towards Integration of CAx Systems and a Multiple-View Product Model ler in Mechanical Design. Acta Polytechnica, 2005, 45 (3), pp.26-31. hal-02291573

HAL Id: hal-02291573

<https://hal.science/hal-02291573>

Submitted on 19 Sep 2019

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.

Towards Integration of CAx Systems and a Multiple-View Product Modeller in Mechanical Design

H. Song, B. Eynard, P. Lafon, L. Roucoules

This paper deals with the development of an integration framework and its implementation for the connexion of CAx systems and multiple-view product modelling. The integration framework is presented regarding its conceptual level and the implementation level is described currently with the connexion of a functional modeller, a multiple-view product modeller, an optimisation module and a CAD system. The integration between the multiple-view product modeller and CATIA V5 based on the STEP standard is described in detail. Finally, the presented works are discussed and future research developments are suggested.

Keywords: concurrent engineering, integrated design, CAx system, product modelling, STEP.

1 Introduction

The demand for a high quality, low cost product with a short development time for the dynamic global market has forced researchers and industries to focus on various effective product development strategies. Concurrent engineering is an effective approach to reduce product development lead-time and improve the overall lifecycle quality by incorporating downstream product development considerations into the early design stage. The product development process deals with various kinds of data issued from value engineering, structural breakdown, DFX assessments, etc. Integration of product data in each design phase (conceptual design, embodiment design and detail design, Pahl and Beitz [1]) will provide more complete information-based decision making. Product modelling has been recognized as an effective technique for facilitating the representation and management of product data.

During the 1990s many research works dealing with product modelling were published by Krause et al. [2], Laakko and Mäntylä [3], Anderl and Mendgen [4]. The main approach of these works was to describe the product based on geometric and form features. In recent years, the research focus of product modelling has gradually shifted to the earlier design phase. Essentially, the earlier design phase is function-driven or function-oriented, because the main design focus at this stage is to find a design solution that is able to achieve the required functions. Form feature modelling approaches and systems, dealing mainly with geometry, cannot take functional information into account at all, because this information is a more abstract concept than geometric and topological information. In such a situation, product modelling oriented to functional design has been widely studied. The main idea of these modelling approaches is to describe the relationships between functions, behaviours and structures during the design process, Umeda et al. [5], Deng et al. [6], Zhang et al. [7].

The representation of the data that has to be managed during the product development process depends on the design phase. It is hardly feasible to find an all-round perfect product model which is able to handle all the product data aspects above. In the presented research work, two models are used to deal with product data from the conceptual phase

to the detail design phase. A Function-Behaviour-Structure (FBS) model is employed for the conceptual and embodiment design phase, and a multiple-view model is used for the embodiment and detail design phase. As we know, commercial CAD systems provide powerful capability in CAD modelling and analysis. Currently, the research aim is to obtain the linking which will integrate these three models.

This research proposes an integration framework about different aspects of product data. On the conceptual level, the different kinds of product data and their logical relations are described through functional modelling and multiple-view modelling and CAD modelling. On the implementation level, the integration of the functional modeller, the multiple view product modeller and CATIA compose the environment for implementing the modelling logics provided on the conceptual level. The data exchange between CATIA and an already-existing optimisation module is also considered to aid designers for a quick assessment of a sub-assembly taking into account a set of evaluation criteria.

This paper is organised as follows: (1) an integration framework for multiple aspects of product data is proposed, (2) as an application of this framework, the FBS model and multiple-view model are briefly described and the linking between them is emphasised, (3) the integration of the implementing modellers is described, especially details of the linking between the multiple-view modeller and CATIA is discussed, (5) the conclusions and future work are presented.

2 The proposed integration framework

The product development process deals with different kinds of data in each design phase. During the phase of task clarification, the design specifications originating from customer requirements are the description of a product to be designed; during the conceptual design, the means (principle solutions or components) are generated to meet the functional requirements; then the configurations of each of the components and the connections between the components are set during the embodiment design; after that, final decisions on dimensions, arrangement, shapes of individual

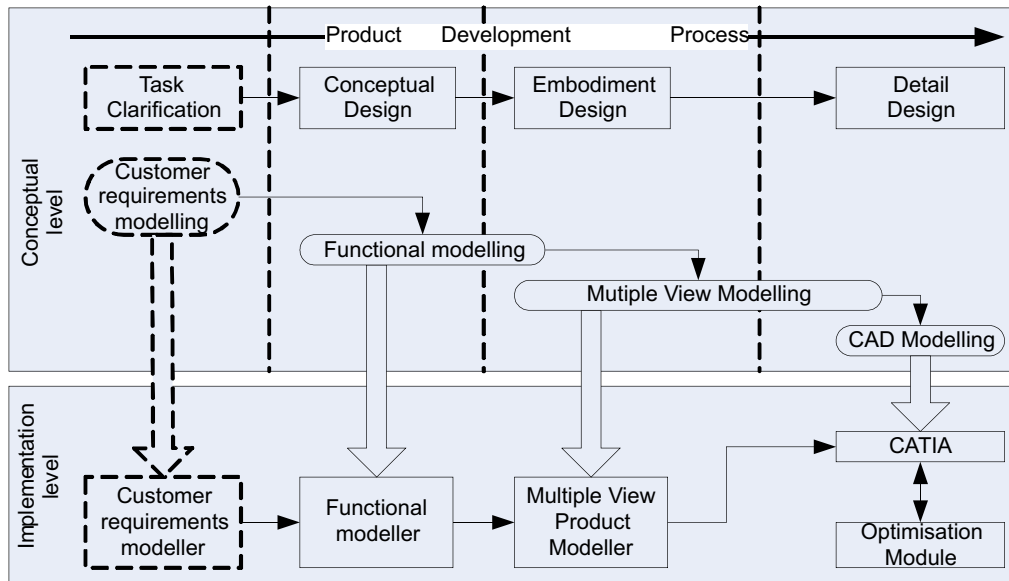


Fig. 1: Overview of the integration framework

components and materials are made, giving due consideration to the manufacturing function in the detail design. As mentioned in section 1, the modelling of product data depends on the design phase. It is obviously difficult to handle all the aspects of product data given above in an all-round model. The key to product modelling is product model partition and integration. Fig. 1 shows an overview of the proposed integration framework. The purpose of this framework is to provide methods and tools to enable product data integration in a concurrent engineering approach.

Several mapping methods have been proposed to implement system integration within the product modelling area. In Nowak et al. [8], a meta-modelling based method has been proposed for the integration of product models. A formal modelling notation is required for the definition of mappings on the conceptual level. This notation provides a method to describe the correspondences between the models. After the mapping has been defined, it is possible to translate data to the implementation level.

According to the identified product data in different design phases, the following modelling techniques are adopted on the conceptual level:

- 1) Customer requirement modelling: the customer requirements are analysed and translated into a statement which defines the function that the product should provide (referred to as a functional requirement) and the physical requirements that the product must satisfy.
- 2) Functional modelling: to describe a design and its requirements from its functional aspects so as to allow reasoning about its functionality and generating schemes.
- 3) Multiple-view modelling: to embed engineering knowledge (geometry, process planning, manufacturing, assembly, etc.) into the description of the form features.
- 4) CAD modelling: to describe the geometric and topological information and to enable an analysis.

According to the product modelling techniques provided on the conceptual level, the corresponding implementing systems have already been developed by research institutes

or commercial companies. These implementing tools and their relationships compose the implementation level in the framework. There are a number of techniques for customer requirement modelling. The most well-known of these techniques is Quality Function Deployment (QFD). The functional modeller and the multiple-view modeller have already been developed respectively by Song [9], Roucoules and Tichkiewitch [10]. The commercial CATIA V5, from Dassault Systemes (www.3ds.com), is used here for classical CAD modelling. To aid designers to make a quick assess went of a sub-assembly taking into account a set of evaluation criteria, the already-developed optimisation module is also considered to be integrated into the proposed framework.

As this research work focuses on mechanical design, customer requirement modelling and its modeller system is not considered currently. On the conceptual level, the integration of functional modelling and multiple-view modelling will be emphasised; on the implementation level, the data exchange between the multiple-view modeller and CATIA V5 will be focused on.

3 Product modelling

In this section, we will first briefly describe the two models our work is based on. Then the integration of these two models will be emphasised.

3.1 Functional modelling

The idea of functional modelling in conceptual and embodiment design is to reason at the functional level in order to generate solutions to specified design problems. The product requirements are progressively mapped onto product structures via functional modelling.

To support functional modelling, it is now generally accepted that design information should include not only the physical structure of a design, but also its required functions and implementing behaviour [6]. For example, the function of a steam valve in a boiler is to prevent an explosion; its behaviour is that it opens when a certain pressure difference is

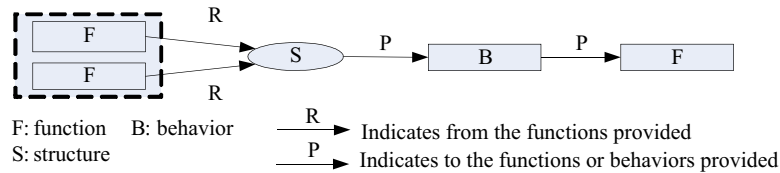


Fig. 2: Functional representation of a structure

detected; its structure is the physical layout and the connection between the various physical components [11]. Fig. 2 shows a functional representation of the structure. This representation suggests that each given structure be described by the behaviours it can provide and the functions provided by the behaviours, and in order to provide each of these, the functions that the structure requires.

The search for a solution to a given problem, defined in terms of a given set of desired functions, would start with a search among the known behaviours for those that can provide the desired functions, then among the known structures for those that can provide the intended behaviours. These chosen structures, in turn, require some other functions in order that they can provide the desired functions. Now, new behaviours and structures will be searched for, so as to provide these required functions, which in turn give rise to new functional requirements. This process will continue until all the required functions are provided by some structures. Each resulting combination of structures evolved by the above process thereby becomes a solution to the design problem posed at the beginning.

As the design level decreases (from abstract to concrete), the difference between the meaning of function and behaviour becomes more and more vague. In such a situation, a function may be mapped onto a structure without the behaviour as the transition. For example, the intended function “transmitting torque” can be mapped onto a structure “shaft”. When a function may not be mapped onto a behaviour or a structure, it may be broken down into sub-functions. Thus, the behaviour or structure in the above representation may be a virtual node. A detailed discussion about the mappings in functional modelling is presented by Song and Lin [12].

3.2 Multiple-view modelling

For embodiment and detail design, a multiple-view model is used. The product has several breakdowns according to different points of view, i.e., the product has multiple-view breakdowns. The multiple-view breakdown was introduced by Chapa and detailed in [10] to gather all the data that describe

a product with a specific vision of it. Fig. 3 shows an example of the multiple-view approach with: technology, manufacturing, assembly and finite element analysis views. Roucoules et al. [13] describe the integration of the process planning view.

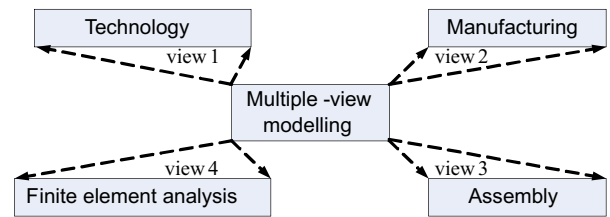


Fig. 3: Multiple-view model

This representation allows multiple-views breakdown of the product and ensures the link with the geometry. These feature-based breakdowns complete the product definition, adding new data and new constraints from specific points of view such as tooling, finite element analysis, etc. The multiple-view model is fully described in [14].

3.3 Integration of the functional model and the multiple-view model (conceptual level)

To integrate the functional model and the multiple-view model, it is necessary to establish the linking between these two models. A feasible way is to find the relationship between the results of functional modelling and the initial information of multiple-view modelling. The scheme generated from functional modelling is described by the information on function, behaviour and structure in terms of the representation model as shown in Fig. 2. The multiple-view breakdowns start from some structures in multiple-view modelling. Thus, both models hold the information on the structural description. It is obvious that the structural descriptions in these two models are in fact different. The objective of functional modelling is to generate an outline scheme, so the structures here are the components on the upper design level. In the multiple-view model, the structure descriptions are relatively detailed. It is nevertheless suggested that the terminal point of functional

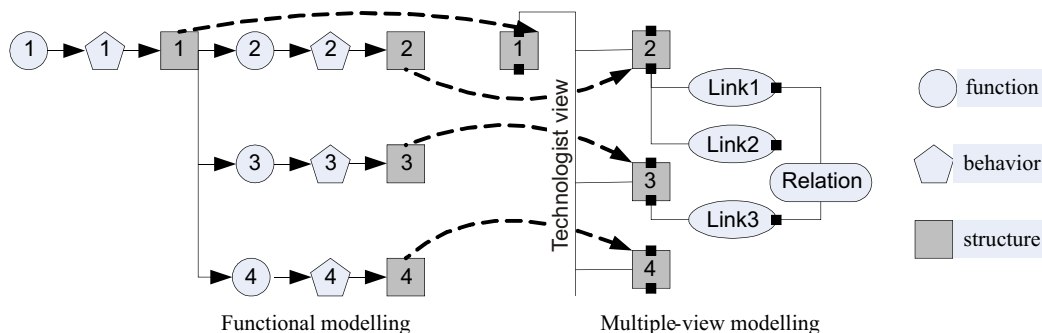


Fig. 4: Integration of the functional model and the multiple-view model

modelling overlaps with the starting point of multiple-view modelling. Fig. 4 shows the integration framework between these two models.

From the integration framework, the structure which is the terminal node in functional modelling can be shifted to the technology view in multiple-view modelling. Thus, functional relationships that exist between structures, such as drive, support, hold, locate, and couple etc., can be analysed further. Figure 5 shows the integration of the two models with an example of a component in a sewing machine.

f1: join-material *f21: tense-stitch*
b1: sewing *f22: catch-thread-loop*
f2: interweave-thread-material *f23: lead-thread-through-material*
b3: thread-takeup *s9: thread-takeup-bar*
s8: needle-bar *s11: brace*

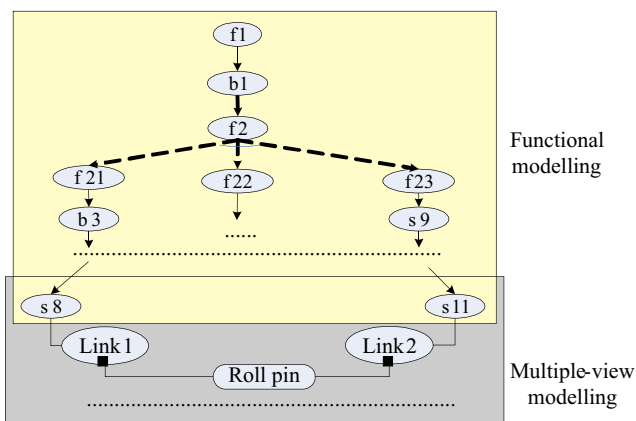


Fig. 5: A case of integration of a functional model and a multiple-view model

4 Integration on the implementation level

In this section, integration between the implementing systems is discussed. There are four different approaches to the data exchange problem: manual re-input of data, direct translation, neutral format translation, and a shared product database [15]. Each approach has its own pros and cons. Direct translation is the simple and accurate solution, but the number of the translators increases exponentially with the number of systems involved. The last two solutions are reasonable, flexible, and adaptable.

4.1 Linking between a functional modeller and a multiple-view modeller

The functional modeller is a knowledge-based expert system, which supports functional reasoning based on domain-specific knowledge. It was developed using a CLIPS expert system shell (C Language Integrated Production System) [16]. As explained in section 3.3, the structural information plays the internuncial role between the functional modelling and multiple-view modelling. After the functional modelling, the schemes which contextualised the functional product data are output in a neutral plain text file. Then the multiple-view modeller can read the related structures from

the neutral file. Thus, according to the structures, the different aspects of the product data can easily be retrieved.

4.2 Linking between CATIA V5 and optimisation module

Integration of the optimisation module into the proposed framework allows designers to quickly assess a sub-assembly, taking into account a set of evaluation criteria and to evaluate the influence of the data on the design solutions by modifying one or more item [17]. The mechanism of data exchange between CATIA V5 and the developed optimisation module is shown in Fig. 6. In CATIA V5, the optimisation problem is abstracted from the parameterised geometric model. The parameters and constraints which define the optimisation problem are output into a neutral file by calling CAA-API (Component Application Architecture Application Programming Interface) of CATIA V5. Then, the optimisation module reads the data from the neutral file and performs the optimisation operation. The optimal values of the variables are output into a neutral file. Finally, CATIA V5 gets back the results from the neutral file by calling CAA-API and updates the geometric model.

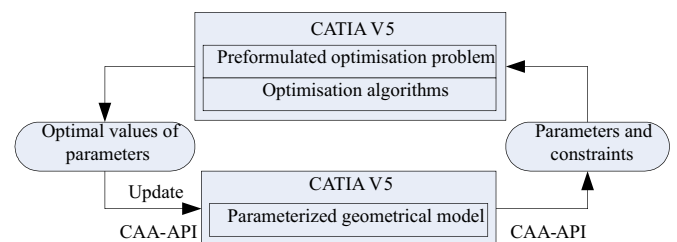


Fig. 6: Data exchange between CATIA V5 and the optimisation module

4.3 Linking between the multiple-view modeller and CATIA V5

In the CAD/CAM context, there exist several standards for data exchange, such as IGES, SET, VDA-FS, EDIF, etc. [18]. The most popular exchange standard in use is IGES. It was designed as a neutral format for the IGES exchange of CAD data, and has been used as the standard for geometric data by most CAD/CAM systems. Although IGES is best supported as an interchange format for geometric data, it cannot fulfill the completeness requirement in representing product data. STEP was first proposed in 1984 [19] to represent complete information of a product throughout its life cycle. Then the choice was made that the data exchange between the multiple-view modeller and CATIA V5 would be based on STEP. Another approach was developed in Eynard et al. [20], aiming at a direct connexion of the multiple-view modeller and an opensource CAD system based on on OpenCASCADE libraries.

The relevant output data from the multiple-view modeller is converted into STEP file format by a translator, and then the STEP file can be opened by CATIA V5 directly. The key to this data exchange is the development of a Translator. This module has already been developed using MS Visual C++ 6.0. Considering the reusability and extensibility of the module, the mapping about an entity from the output file of the multiple-view modeller onto the STEP neutral file is

encapsulated into a class which provides a compact interface for accessing. When an entity of the STEP file is needed, the only thing to be done is to extract the required parameters and call the corresponding class member function. For example, the cylindrical surface defined by the multiple-view modeller is shown as follows:

```
{ axe SLine : pl_axe_doigt_cyl4
    dir Vector : pl_dir_axe_doigt_cyl4
        z Float : [0.000000..0.000000]
        y Float : [1.000000..1.000000]
        x Float : [0.000000..0.000000]
    diameter Float : [12.0 .. 12.0]
    position Point : pl_pos_doigt_cyl4
        z Float : [0.000000..0.000000]
        y Float : [ 20.000000.. 20.000000]
        x Float : [0.000000..0.000000]
    length Float : [30.0 .. 30.0]}
```

The parameters required by the translator are the center points of each section circle and the radius. The two center points are center 1 (0, 20, 0) and center 2 (0, 10, 0); the radius is 12. Then, a cylindrical surface object (css) can be built with the parameters and we call its member function (GetSTEPCode) to get the STEP code.

```
CCylindrical_surface_segment css(center1,center2, radius),
    Css.GetSTEPCode.
```

The obtained STEP file is shown as follows (in part):

```
#51=CARTESIAN_POINT('Axis2P3D Location',( 5.,0.,0.));
#52=DIRECTION('Axis2P3D Direction',(0.,1.,0.));
#53=DIRECTION('Axis2P3D XDirection',( 1., 0.,0.));
#54=AXIS2_PLACEMENT_3D('Cylinder Axis2P3D',
                        #51,#52,#53);
#55=CYLINDRICAL_SURFACE('generated cylinder',
                        #54,6.); .....
```

Fig. 7 shows the interfaces of the process of data exchange.

5 Conclusion

The research work presented in this paper deals with product data integration in design. Lasts of this problem have been studied in the literature and some concepts have been well specified. The current research work focuses on developing a new integration approach and associated systems.

An integration framework of a CAx system and a multiple-view modeller is proposed. On the conceptual level, an FBS model and the multiple-view model are used to represent the product data during the design process. The linking between these two models is established to enable the integration of product data in each design phase (conceptual design, embodiment design and detail design). On the implementation level, the data exchange between the corresponding modellers is carried out by neutral file translation. Therefore, the designer can easily access the product data from different views and contextualise it throughout the product development process. This research work provides designers with a comprehensive product model and their modelers so as to help them to make information-based design decisions.

References

- [1] Pahl, G., Beitz W.: *Engineering Design : A Systematic Approach*. Springer. London, 1996.
- [2] Krause, F. L. et al: "Product Modelling." *Annals of the CIRP*, Vol. 42(1993), No. 2, p. 695–706.
- [3] Laakko, T., Mäntylä, M.: "Feature Modelling by Incremental Feature Recognition." *Computer-Aided Design*, Vol. 25 (1993), No. 8, p. 479–492.
- [4] Anderl, R., Mendgen, R.: "Modelling with Constraints: Theoretical Foundation and Applications." *Computer-Aided Design*, Vol. 28 (1996), No. 3, p. 155–166.
- [5] Umeda, Y., Ishii, M., Yoshioka, M., Tomiyama, T.: "Supporting Conceptual Design Based on the Function-Behavior-State Modeler." *Artificial Intelligence for Engi-*

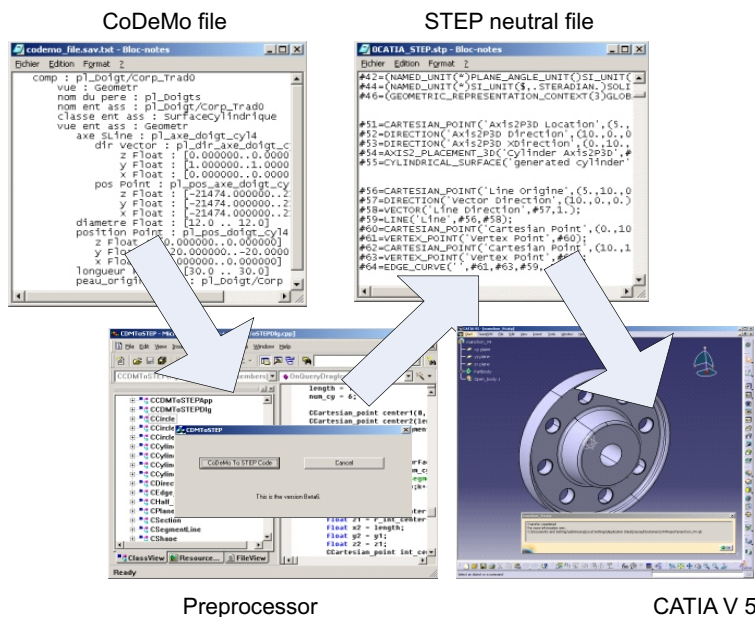


Fig. 7: Example of data exchange between a multiple-views modeller and CATIA V5

- neering Design, Analysis and Manufacturing, Vol. **10** (1996), p. 275–288.
- [6] Deng, Y.-M., Tor, S. B., Britton, G. A.: “A Dual-Stage Functional Modelling Framework with Multilevel Design Knowledge for Conceptual Mechanical Design.” *Journal of Engineering Design*, Vol. **11**(2000), No. 4, p. 347–375.
- [7] Zhang, Tor, S. B., Britton, G. A., Deng Y.-M.: “EFDEX: A Knowledge-Based Expert System for Functional Design of Engineering Systems.” *Engineering with Computers*, Vol. **17** (2001), No. 4, p. 339–353.
- [8] Nowak, P., Roucoules, L., Eynard, B.: “Product Meta-Modelling: An Approach for Linking Product Models.” In: Proc. of the IEEE World Congress on System, Man and Cybernetics. Tunisia: Hammamet, October 6th–9th, 2002.
- [9] Song, H. J.: “Research on Methodology and Key Techniques of Scheme Generation in Conceptual Design of Mechanical Products.” PhD Thesis (in Chinese). Xi’an Jiaotong University, China, 2003.
- [10] Roucoules, L., Tichkiewitch, S.: “CoDE: a Co-operative Design Environment. A New Generation of CAD Systems.” *Concurrent Engineering Research and Application*, Vol. **8** (2000), No. 4, p. 263–280.
- [11] Kuipers, B.: “Commonsense Reasoning About Causality: Deriving Behaviour From Structure.” *Artificial Intelligence*, Vol. **24**(1984), No. 1–3, p. 169–203.
- [12] Song, H. J., Lin, Z. H.: “Hierarchical Function Solving Framework with Hybrid Mappings in Conceptual Design of Mechanical Products.” *Chinese Journal of Mechanical Engineering*, Vol. **38** (2003), 5, p. 82–87.
- [13] Roucoules, L., Salomon, O., Paris, H.: “Process Planning as an Integration of Knowledge in the Detailed Design Phase.” *International Journal of Computer Integrated Manufacturing*, Vol. **16** (2003), No. 1, p. 25–37.
- [14] Tichkiewitch, S.: “Specification on Integrated Design Methodology Using a Multi-view Product Model.” In: Proc. of ASME Engineering System Design and Analysis Conference Montpellier, France, July, 1996.
- [15] Fowler, J.: *STEP for Data Management, Exchange and Sharing*. Technology Appraisals, 1995.
- [16] Giarratano, J., Riley, G.: *Expert Systems: Principles and Programming*. 3rd ed. Boston, PWS, 1998.
- [17] Eynard, B., Lafon, P.: Towards a Mechanical Systems Modelling and Optimal Embodiment Method. In: Proc. of 13th International Conference on Engineering Design Glasgow, Scotland, UK, August 21st–23rd, 2001.
- [18] Bloor, M. S., Owen, J.: “CAD/CAM Product-Data Exchange: the Next Step.” *Computer-aided Design*, Vol. (1991), p. 237–243.
- [19] ISO10303-1. STEP: *Industrial Automation Systems and Integration Product Data Representation and Exchange, Part 1*. Overview and fundamental principles, 1994.
- [20] Eynard, B., Roucoules, L., Yan, X.T.: “Knowledge Integration Approach for Product Modelling Using an Opensource CAD System.” In: Proc. of 5th Knowledge Intensive CAD IFIP5.2 Workshop. St. Julian, Malta, July 23rd–25th, 2002.

Dr. Huijun Song
 phone: +33 325 715 671
 fax: +33 325 715 675
 e-mail: song.huijun@utt.fr

Dr. Benoît Eynard
 e-mail: benoit.eynard@utt.fr

Dr. Pascal Lafon
 e-mail: pascal.lafon@utt.fr

Dr. Lionel Roucoules
 e-mail: lionel.roucoules@utt.fr

Troyes University of Technology
 Laboratory of Mechanical Systems and Concurrent Engineering–FRE 2719 CNRS
 12 rue Marie Curie
 BP 2060, F.10010 Troyes Cedex, France