A New Method of Visualization and Documentation of Parametric Information of 3D CAD Models

Maxim Marchenko[1], Bernd-Arno Behrens[2], Gregor Wrobel[3], Robert Scheffler[4] and Matthias Pleßow[5]

[1]Leibniz Universität Hannover, marchenko@ifum.uni-hannover.de
[2]Leibniz Universität Hannover, behrens@ifum.uni-hannover.de
[3]Society for the Promotion of Applied Computer Science, wrobel@gfai.de
[4]Society for the Promotion of Applied Computer Science, scheffler@gfai.de
[5]Society for the Promotion of Applied Computer Science, plessow@gfai.de

ABSTRACT

The modeling process of complex assemblies with modern parametric 3D CAD systems generates a large network of various parametric dependencies and parameter information that is difficult to manage. In this paper, a new method and a software solution for extraction and visualization of all parametric information within 3D CAD models are presented. The requirements for the visualization of complex 3D CAD models are discussed and an overall solution concept is described. Based on the general structural analysis of the parametric 3D models, a CAD system-independent graphical language with two forms of visualization will be introduced. A data model to store parametric information from CAD models, as well as a method to export them has been developed. This paper also presents the implementation of the method in CATIA V5 and the validation on a number of CAD models.

Keywords: parametric modeling, schematic representation, sheet metal forming.
DOI: 10.3722/cadaps.2011.435-448

1 INTRODUCTION

Computer aided design (CAD) as one of the first application domains for computer aided engineering (CAE) is an essential tool for many disciplines. Since the end of the last century, complex objects could only be represented virtually because more powerful software was available. This was the basis for many innovations.

In the 1980s schematic diagrams started to show up as a computer aided application next to CAD. The geometry of the constructed objects isn't the focus here, but rather the modeling of the human decision-making processes or the structural and behavioral models of the examined objects. These models can reach a high degree of sophistication and are often structured in a network that contains hierarchic relations. Working with these models requires special CAx tools that are conceptually different CAD applications. Moreover visualizations have to be tailored for specific purposes.

Over the last decade, the use of modern parametric 3D CAD systems has steadily progressed. These systems control the model modification process through variable parameters. A characteristic feature is the definition of relations between parameters, geometric elements, assemblies etc. [1, 2]. Not only beginners, but also experienced designers, face the challenge of effectively using all the features available in parametric 3D CAD systems. The modeling of complex assemblies requires an enormous network of parametric information and parameter dependencies. These may be complicated and confusing even for experienced designers. An unfamiliar user can be completely hampered [3, 4]. That’s why many companies, even those using the parametric methods of modern 3D CAD systems, have difficulties with documenting the inherent design knowledge and visualizing the parametric information [5]. Current forms of documentation, like technical drawings and standard CAD exchange formats (e.g. STEP, IGES), don’t contain representations of parametric information [6, 7, 8]. However, there is a very high demand to exchange the history-based parametric models among different CAD systems. The actual investigations of the ISO 10303 (STEP) show the possibility to exchange construction history, parameters, constraints, features and other elements of "design intent" present in the model [9, 10]. Especially three fundamental outlines define the exchange of parametric-associative 3D CAD models: part 55 represents the sequencing of design operations, part 108 is responsible for representation for the association of explicitly defined parameters, definition of geometric constraints and description of 2D sketches and part 111 defines the core feature types. However, the STEP format does not support the visualization of these elements.

There is no tool available yet, that can automatically visualize all the relations in a parametric 3D CAD model for documentation. Most tools offer only tree views to navigate in the model elements that hamper the understanding of complex relations. This paper presents novel visualization concepts, which facilitate both the management and the understanding of logical information in parametric CAD models. Further, a prototypical implementation is presented along with examples of achieved results. To narrow the scope of the project, the focus lies primarily on the design of sheet metal forming tools. Due to the diverse range of variants within this sector, parametric systems are widely used. Parametric design has proved its capabilities in this branch. However, the developed visualizations and tools can be generalized for design tasks in other branches.

2 DEVELOPMENT METHODS AND CAD IN SHEET METAL TOOL DESIGN

Different fundamental design methods are established as an important basis for computer aided design, so that structured approaches for product engineering are provided. In the case of product design and development, the difference is clarified between science and design methodologies [11]. As part of the science methodologies, i.e. applying scientific methods, technical systems are analysed with respect to its structure and its relationship to the environment, so that procedures for development can be derived from detected interrelationships. Essential content of the design methodology is to formulate concrete guidelines for the development and design of technical systems. Important approaches of design methodology were provided by several authors, e.g. Pahl and Beitz [11], which are also included in the VDI Guideline 2221.

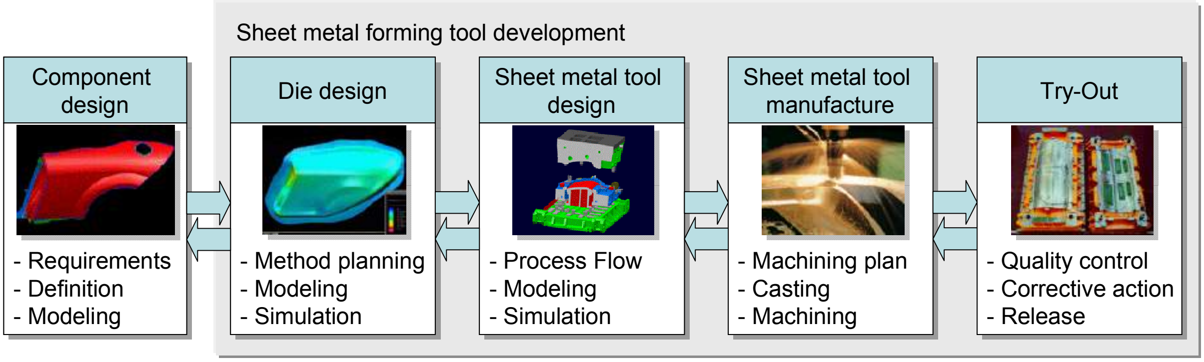

Fig. 1: Steps in sheet metal forming tool development [10].

The design of a mechanical component in sheet metal forming tools has a number of development steps that occasionally overlap (Fig. 1). At the beginning of the tool development process the definition and the modeling of the sheet metal component are performed. This step also includes the feasibility study and the comparison of state of the art technologies, the definition of the tasks as well as preparation of the requirement list. The die design stage contains the definition of operation sequences, the creation of necessary tool surfaces and primary simulations. The step focuses on surface modeling and the definition of elements that are directly dependent on the component geometry [12]. The primary goal is the minimization of the number of required tools at high operating reliability. The tools are simplified in the draft, in close cooperation between component design and production departments to combine multiple forming and shearing operations at the manufacturing station. However, the more operations are integrated into one forming tool, the more complex the tool structure will be. The next step contains the modeling of the sheet metal tooling system. In general, the design is based on internal company rules, which are defined in standardized catalogues, for example wall and ribs thicknesses. Many forming tools must be designed so that they can be used in various types of presses. Master design models or standardized templates can help here [13]. The subsequent steps casting, machining and try-out do not really influence the component design but are more oriented toward production. The arrows in Fig. 1 indicate that there are iterations between different steps, particularly between component design and die design because of the simulation that validates the component geometry [12].

The excursus on design and development of sheet metal forming tools at this point illustrates how complex the processes in modern parametric CAD systems can be. Throughout the whole product development process of forming tools, the highest possible parallelization of process steps is required, which have to be parameterized and mapped into the CAD system. In addition the designer usually has to manage very complex models and hence needs supporting software tools, which help him to deal with the complexity of the described design and development processes in collaboration with other engineers.

3 PARAMETRIC GEOMETRY MODELING

3.1 Definitions

Modern parametric 3D CAD systems, like CATIA V5 or SolidWorks from Dassault Systémes, Unigraphics NX from Unigraphics Solutions or Pro/ENGINEER from Parametric Technology Corporation, allow simultaneous use of the three basic elements in parametric geometry modeling:

1. parameters,
2. features (design elements) and
3. their dependencies (relations and constraints).

The **parameters** allow the use of variable or constant values to define properties and dependencies within and between parts and assemblies. The relevant literature [1, 14, 15] offers various classifications of parameters. In general, these can be divided into geometric and non-geometric (functional) parameters. A geometric parameter includes a value and a dimension. The non-geometric parameters usually include technological properties like material or strength and can also be used for organizational issues

The **features** are the second part of the parametric geometry modeling. A feature can be defined as an object in an information system (such as a CAD-system), which has geometric and non-geometric parameters as well as methods to describe its properties and to change them, resulting in the semantics of the object [16, 17].

The **dependencies** such as global constraints and boundary conditions of geometric and non-geometric parameters describe relations between the parameters and CAD elements, such as features, parts or assemblies [2]. The dependencies enable the modeling of functions and equation systems as well as algebraic, logical and semantic constraints. In general, dependencies are very similar to programming languages and can be realized as a description of context-free grammar.

Generally, parametric geometry modeling can be described by specifying the parameters, which have a name and a value, in opposition to explicit geometric modeling, in which the changes in the model are only possible by redefining the entire geometry.

3.2 Structure of Parametric 3D CAD Models

Parametric 3D CAD models are characterized by a hierarchical structure. It consists of general assemblies, subassemblies, parts, features, dependencies and parameters [18]:

- Assemblies (AS) are on the highest level of the product structure. They are never children of other model elements, but they may have other elements as possible children.
- Subassemblies (SA) can only be children of assemblies or other subassemblies. This enables the creation of subassemblic hierarchies.
- Parts (PA) occur either individually or as children of assemblies and subassemblies and may have other elements as children.
- Features (F) are children of assemblies, subassemblies or parts and have the dependencies and parameters as children. In addition, parent-child relationships between features are possible.
- Dependencies (D) can be children of assemblies, subassemblies, parts or features and have parameters as children.
- Parameters (P) can parameterize all the CAD elements and can therefore also be children of all other CAD elements.

Generally, these elements build a monohierarchical, treelike structure, which is usually not very deep, but has a relatively broad expansion. In this case, all elements stand in relation to exactly one parent. This is called the model structure tree.

If the structure also contains additional dependencies between individual CAD elements on separate hierarchical layers, then the result is a directed acyclic graph, also called polyhierarchy, in which multiple superior elements are possible. A very complex network structure will be produced from a manageable tree structure of the CAD model.

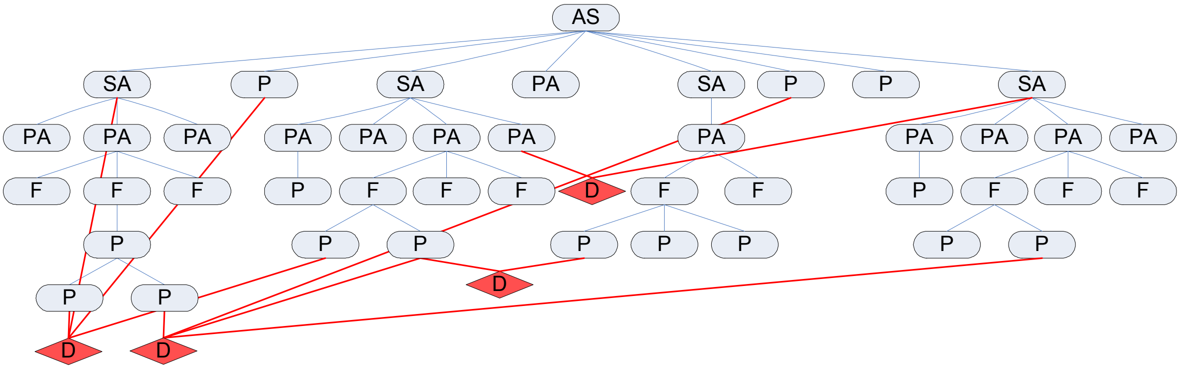

Fig. 2: Polyhierarchical structure.

In Fig. 2 the model structure tree with additional dependencies is expanded to a polyhierarchical structure (model structure graph). Generally, in parametric CAD systems, the number of such dependencies is not limited. Furthermore, these can also vary in complexity, for example a dependency can include a simple formula or a complex macro.

3.3 Problems with Existing Visualization Concepts of CAD Systems

The designed geometry, parts and assemblies in modern parametric CAD systems are listed chronologically in a tree structure (examples for this are the CATIA V5 specification tree and the Pro / ENGINEER model tree). For parts or small assemblies, this tree structure gives an adequate overview of the design elements and the complete model. However, analysis and understanding of complex relationships in very large models are only partially possible, using a tree structure.

Alternative visualization techniques that exist in addition to the above tree representations, for example the display of parent-child relationships, such as CATIA V5 (Fig. 3), have problems in visualizing parametric-associative 3D CAD models: When multiple CAD elements are shown, the links between the CAD elements often can not be drawn without intersections, resulting in a confusing visualization. Furthermore, the hierarchy of individual CAD elements often cannot be detected (e.g. *Pad* in Fig. 3) and the designation of certain parameters is not unique, for example, the features *Pad* or *EdgeFillet* in Fig. 3 appear more than once. Another very important property that is not shown in these images is the representation of input and output parameters, which hampers the tracking of information flows in the parameter network. Finally, only icons are used to distinguish the individual CAD elements. This is often very confusing due to the abundance of existing CAD elements and when the designer switches between different CAD systems, he has to become accustomed to the new visualization.

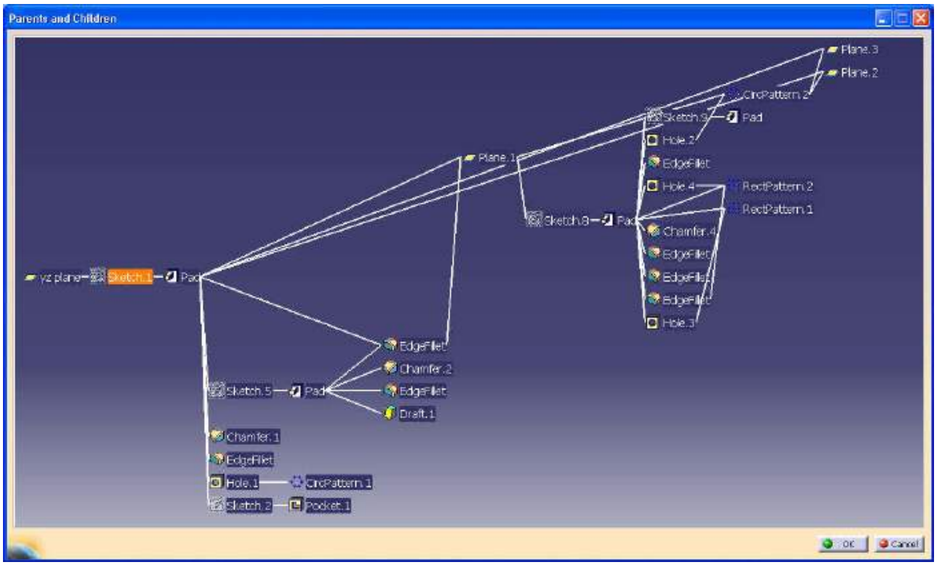

Fig. 3: Problems with visualization of parent-child relationship in CATIA V5.

4 REQUIREMENTS FOR THE VISUALIZATION OF 3D CAD MODEL STRUCTURE

There are two fundamental subtasks in order to visualize the structure of parametric 3D CAD models: The CAD elements have to be distinctly represented on the screen and the graphical symbols should be neatly arranged.

The representation of CAD elements is described by a *graphical language* consisting of symbols and rules on the placement and connection of these symbols. That means each element in CAD models has to be provided with a graphical symbol. The symbols must differ to increase the clarity and unambiguousness of the visualization. The tools to achieve this goal are shapes, colors and icons.

The primary goal in finding a *layout* for the graphical language is readability of the visualization. The aesthetic criteria required for that are: clear boundaries between graphical elements, objects and lines should be drawn without overlapping. Connections represented by lines should be easy to follow (e.g. by minimum crossings and bends).

While these constraints apply to any structure, for a good layout the specific properties of parametric 3D CAD models must be taken into consideration. The hierarchy contained in the model is crucial for this: it's usually broad but not deep. A primary requirement for the layout is therefore the clear, distinguished and understandable display of the hierarchic structure. The position of each element according to the hierarchy should be intuitively understandable. This also helps differentiating between similarly named elements.

Parametric CAD models contain additional dependencies among model elements. These connections can't be displayed clearly by a static visualization, as the sheer number of dependencies would overwhelm the viewer. The designer needs an interactive approach to reduce the number of elements that are displayed at a time.

The reduction of elements can be achieved easily by only showing zoomed parts of the structure. Another approach are filters that hide specific elements or types of elements. Navigation in the model also reduces the amount of displayed elements. The user should be able to show and hide hierarchic levels or follow relations to other parts of the model. All of these methods require specific interactions to be presented to the user.

To ensure a clear presentation, the previously described interactions have to be built with topological stability. Even structural changes (e.g. the showing or hiding of elements) have to keep the general characteristic of the model.

5 NEW METHODS OF VISUALIZATION OF PARAMETRIC 3D CAD MODELS

There are earlier efforts to create better visualization methods for the structure of CAD models. In contrast to the approach in this paper, these focus on early stages of mechanical design. The well-known modeling languages UML and SysML [19, 20, 21] have been used to help designers create new structures and then implement them in a CAD model. The work presented here focuses on the other end of the process. This paper presents a method to extract parametric information from existing CAD models and generate dynamic visualizations of the underlying structure. These visualizations can then be used in model analysis and documentation. Certainly, SysML can be used to visualize such information, but to fulfill the requirements described in section 4 a new graphical language is required. The following sections describe the common features of our visualization methods and the techniques used to present the parametric information in a clear and useful way.

5.1 Graphical Notation

First of all, to make a notation of graphical language, the objects to visualize must be defined. The experience with different 3D CAD models has shown that a classification of CAD elements is needed to fulfil the requirements for the visualization of parametric information. Thus, five types of CAD elements are proposed:

- **Structural Elements** to implement a parent-child relationship that create a hierarchical model,
- **Geometric Elements** that represent 2D geometries as well as complex 3D features,
- **Parameters** which define constant or variable factors of geometric or non-geometric properties within the 3D CAD model,
- **Dependencies** that describe the relations and constraints in a 3D CAD model, in addition to
- **Interfaces** which are used for the exchange of information among the parts or assemblies.

For the graphical notation, the CAD elements are generally divided into two groups: relationship elements and non-relationship elements (model elements). Although there are connections between two model elements, these are visualized by different arrows or additional symbols. Furthermore, small image icons, color schemes and geometric forms help to distinguish and recognize the individual CAD elements.

Structural elements, geometric elements, parameters and interfaces are model elements and they are represented as rectangles containing name and symbol (Fig. 4a). The color of the model element clarifies the type of each element. The color of individual CAD element types should not be set at this point, because the user should be able to set their own color definitions for individual CAD element types.

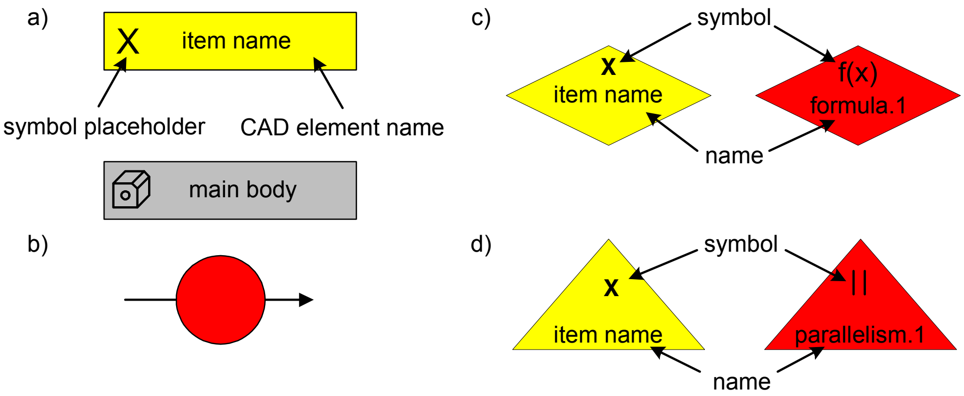

Fig. 4: Templates and examples for the representation of a) CAD elements, b) direct links between CAD elements, c) dependencies and d) constraints.

Relationship elements, in contrast to model elements, have different graphical forms, because they build the polyhierarchical structure of a CAD model and thus are of particular importance for hierarchical discrimination.

In CAD systems, especially in CATIA V5, different dependencies, such as constraints (offset, parallelism, concentricity, etc.) and relations (formula, checks, rules etc.) can be visualized using different icons. This visualization method should also be followed here. However, the identification of a dependency type or any other CAD element, with the help of an icon is often difficult, because remembering and distinguishing a large number of different dependencies in the form of several icons is exhausting for the designer. Furthermore, most icons are CAD system specific, which increase the difficulty level when switching to a new software environment. Therefore, it is necessary at this point to define a general approach that allows identifying the dependencies immediately.

A good approach to represent relationships between two entities depends on the database modelling concept known as entity-relationship diagram. The advantages of this approach include the uniqueness of the entities and relations, the connection of two or more entities through a relation, indicating the direction of the information flow by arrows as well as the building of chains. These entity-relationship features which are used by dependencies in the visualization and layout are shown in a diamond form (Fig. 4c).

In CAD models, particularly in CATIA V5, the constraints are represented by a large set of dependencies. For this reason, it is advantageous to offer the designer a separate presentation for constraints. The constraints in CATIA V5 can be a maximum of three CAD elements in relation. In this case, a triangle form is beneficial for their presentation (Fig. 4d).

Special cases in the visualization of CAD elements can be represented by direct connections, for example the geometric logic of a line spanned by two points. In this case, no real relation element exists between the model elements. Such connections between different model elements are shown by a circle form (Fig. 4b). The visualization of the information flow between the CAD elements is depicted with a directional arrow. Generally, it is also possible to visualize a non-directional connection through a line without an arrow.

5.2 Parameter Tree

The *Parameter Tree* visualization is based on the hierarchic structure of the CAD model. The model elements are represented by stacked rectangles that are indented according to their hierarchical depth. This is a common way of visualizing trees and therefore intuitively usable. It takes advantage of the low depth and high breadth of the underlying structures. The tree extends only in one direction (downwards), so the resulting space can be used to show the additional hierarchy spanning dependencies.

The usual methods of interaction are present in a *Parameter Tree*: Tree elements can be expanded and collapsed; the focus can be moved and zoomed. This keeps the activities of the designer consistent with his experiences from the model tree in CAD tools. To further reduce the amount of displayed elements it's possible to filter them by type.

To visualize the dependencies, the model is displayed in a second tree view on the right side of the screen (Fig. 5). Navigation is independent in the trees. The gap in the middle is used to show the associations between the model elements interactively.

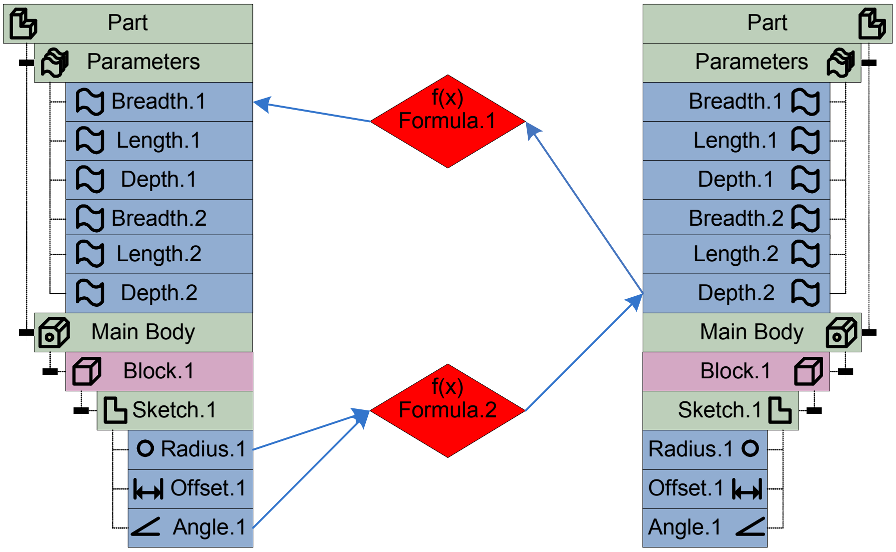

Fig. 5: Parameter Tree showing dependencies between different elements.

By highlighting a model element in one of both trees the user can display its dependencies. An example would be the visualization of all formulas that contain a given parameter. This also highlights all connected parameters in the other tree. Depending on its state of expansion, the connections are drawn to the elements themselves or to the next visible parent element.

The visualization using a double tree with the dependencies in the middle section enables new ways to interact with the model by following paths. The user can switch the focus of the display between the trees and thereby follow a chain of dependencies. The traveled path can be stored and later visualized as an indirect dependency in a *Parameter Plan* (Fig. 6).

The developed Parameter Tree satisfies the stated constraints for the visualization of CAD models. The clear representation of the hierarchy is extended by an easily readable display of the dependencies.

5.3 Parameter Plan

While *Parameter Trees* focus on the hierarchic structure of the CAD models, *Parameter Plans* visualize the mesh of interconnected dependencies. The basic idea is to display a single dependency in the center and place the involved model elements around it. A linear array on the top and bottom of the dependency element proved to be beneficial. The difference between input and output elements is symbolized by directional arrows in the connecting lines.

Contrary to *Parameter Trees*, the elements of this visualization aren't embedded in their hierarchical structure. The symbols for model elements are therefore expanded to display some hierarchical information. Superordinate elements are shown as stacked rectangles to make each element in a *Parameter Plan* uniquely identifiable (Fig. 6). Displaying the hierarchy also allows the visualization of common roots relating the model elements. Since the hierarchic structure is a tree, it's possible to arrange the model elements in such a way that connections between identical hierarchy elements can be drawn as a thick bar. This visual style helps the designer to easily identify the hierarchic bond of different model elements. The four right elements in Fig. 6 (Radius.24, Offset.25, Angle.26, Offset.27) clearly show that they are at the same position in the hierarchical structure of the model due to the thick bars between their respective Sketch.5, Block.4 and Main Body rectangles. This allows to conclude at a glance that Formula 11, the main association in this image, gets input parameters from two distinct places: Sketch.5 and the global parameter list in the model.

Beginning with the displayed model elements, additional associations can be shown. This forms chains that show a clear picture of the indirect dependencies in the CAD model. Smart selection of the displayed connections can make even complicated structures easy to understand.

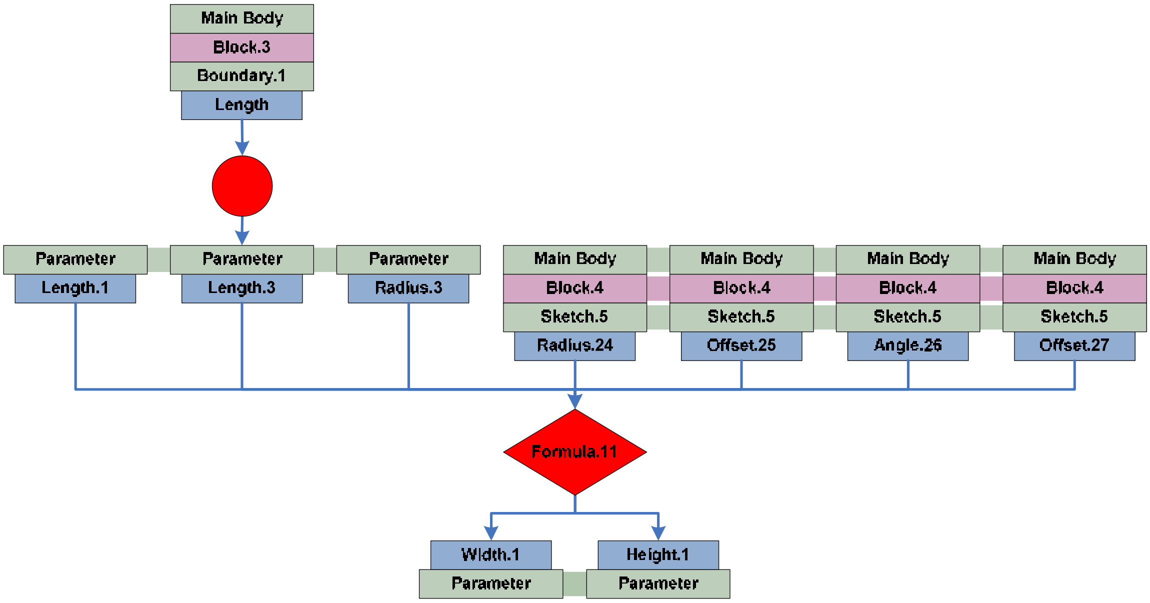

Fig. 6: Example of a *Parameter Plan.*

6 SOFTWARE SOLUTION

6.1 System Architecture

For the documentation and visualization of parametric and associative information, a software concept, whose system architecture is shown in Fig. 7, has been designed. The developed visualization system (VIPA) is modular and consists of three software components: model scanner, layout system and graphical user interface.

The functions of the model scanner are reading, analyzing and storing the CAD model structure and associative information of the CAD model in a CAD system independent data format. Extensible Markup Language (XML) is well suited for storing hierarchically structured data [22]. The layout system interprets the XML file and displays parametric information including dependencies. Through the graphical user interface, the designer can customize the look of the visualization and save it as a project or export it as an image file. The project file is also based on XML and extends the model scanner generated file by the representation of the layout system. The designer can generate specific views of the CAD model and save them inside the project file. The exported images and the project file can be archived and later used for further analysis or documentation.

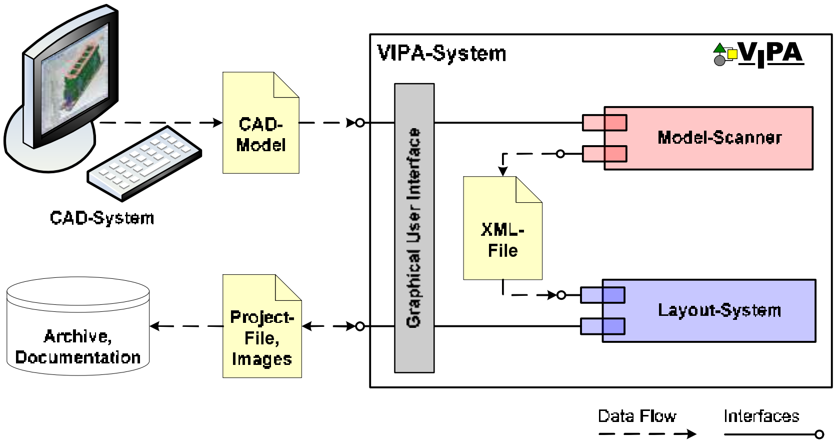

Fig. 7: Architecture of Visualization of Parametric Information (VIPA) System.

The CATIA V5 CAD system was selected for the implementation the VIPA system, because of the access to many tool designs for sheet metal forming at the Institute of Metal Forming and Metal Forming-Machines (IFUM). The visualization concepts and the developed methods can also be implemented for other parametric 3D CAD systems.

6.2 Data Model and Peripheral CAD Interface

It would be possible to implement the ISO 10303 (STEP) format for the extraction, storing and exchange of history-based models [9, 10]. However, the STEP format does not support the visualization of parametric-associative models in a schematic representation. This task requires a custom-tailored data model containing additional information for example regarding graphical elements. Therefore a subset of the data model ELADO [23] was used as a basis for the necessary XML data structure in order to facilitate the implementation of the peripheral CAD interface, including the visualization concepts. ELADO stands for *Extended Layout Data Model* and contains the components depicted within the packages of Fig. 8 as a main property.

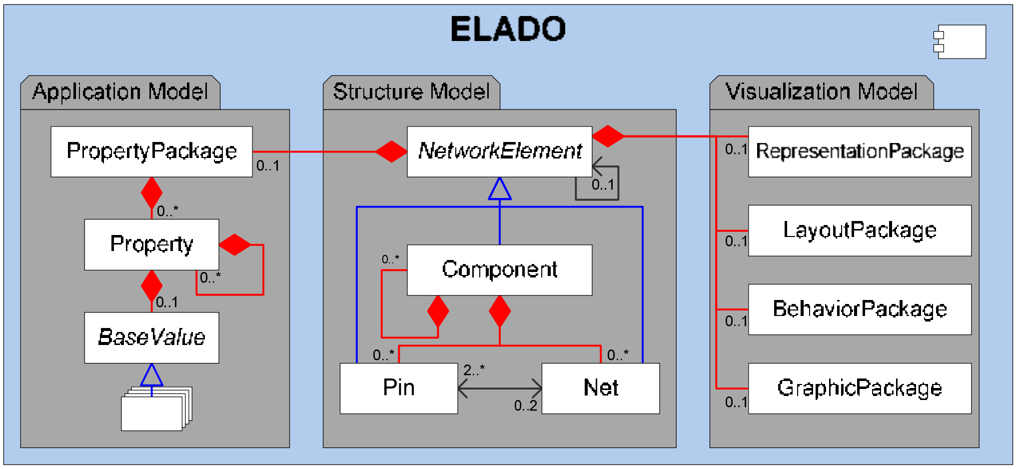

Fig. 8: Main Structure of ELADO Data Model.

The structural model is the central component which allows the illustration of any complex networked structure. The class "NetworkElement" is the root of the class hierarchy and generates the most important characteristics and methods, for example an "ID". The class "Component" forms a container for all elements of a construction tree. The class "Pin" represents the connection points of a component, so that connections (or relations) between components (or CAD elements) can be realized. The class "Net" contains the link between these connection points and represents the actual connection.

The visualization model contains information on various aspects (e.g. structure trees, schematics) in order to graphically represent the structural model. This includes the graphic depiction of the elements in addition to rules for the schematic layout as well as the interactive behavior.

In order to store application specific data (e.g. technology data, business data), ELADO has been augmented with a third model part, the application model. In contrast to the structure model, in which each relevant element is modeled by a class, the application model is an abstract data structure. The data is stored in an associative property tree. Each node in the tree can be identified by a key and owns a value object and a sub tree. In the VIPA system, the name, type and actual content of a CAD element will be saved within a "PropertyPackage", with the aid of the "Property" class. The actual values will be represented by means of the abstract class "BaseValue" that can contain basic types such as Integer, String or Boolean values.

A subset of the ELADO Model (structural and application model) was converted in the model scanner for the implementation of the peripheral CAD interface. The visualization information was not required at the time of the CAD model analysis. In the first step of the analysis of a parametric 3D CAD model, a CAD system specific configuration document which was loaded assigned a unique number to the name of a CAD element. The mapping of a name to a number achieved a CAD system-independent implementation of the software, and as a result, CAD system-independent visualization of parametric information. However, the developer must proceed manually to perform the assignment process just once. No further manual operations are required even if the CAD model was changed. In this project, the mapping was carried out using the CATIA V5 system. In the second step, CATIA V5 was initialized and started with the aid of the CATIA automation interface. The CAD document, defined by user input, is automatically opened and the CAD model is loaded. The opened model is analyzed and the XML data structure is populated with the hierarchical and parametrical information of the model. Furthermore, relationships between separate CAD elements are analyzed and saved in the XML document. The resulting XML document representing the CAD model is then read by the layout system and visualized with the help of the graphical user interface. The XML file documents the structure and the parametric-associative information of the CAD model. It can be archived or exchanged among engineers for further model analysis.

6.3 Software Prototype

The VIPA system shown in Fig. 7 has been implemented prototypically as part of this research project. The prototype is based on the *Framework for Graph-based Engineering Systems (ForGE)*, an advancement of the framework described in [23]. The framework contains several software components that facilitate the fast development in a short time.

Fig. 9 shows two screenshots of the VIPA system. Both images display parts of the CAD model depicted in Fig. 10a. On the left, the VIPA system shows a *Parameter Tree* and a corresponding *Parameter Plan* is shown to the right. Both application windows also contain the hierarchical structure of the model as a tree view.

Using the developed software prototype, CATIA V5 CAD models can be read and visualized as a *Parameter Tree* and multiple optional *Parameter Plans*. Both representations provide the numerous interactions described above. The current display can always be stored for further tasks or exported to an image file.

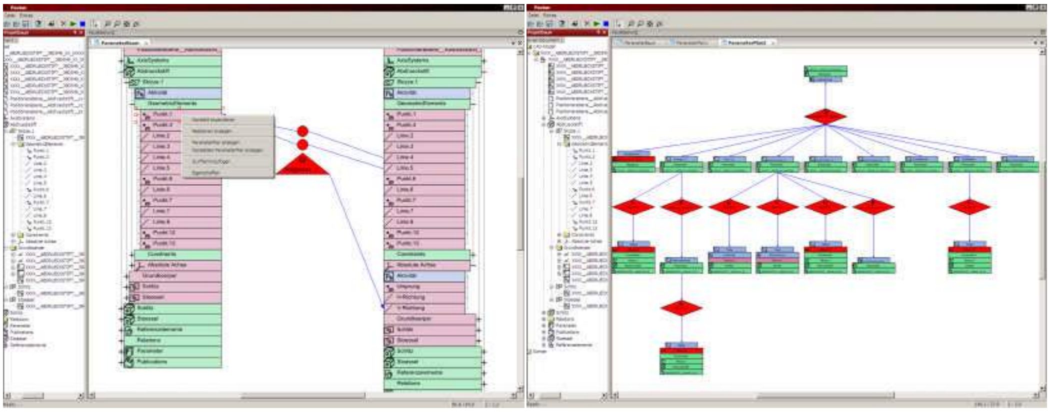

Fig. 9: Screenshots of VIPA software prototype (in German language); left - parameter tree, right - parameter plan.

7 CASE STUDY: ANALYSIS OF SHEET METAL FORMING TOOLS

The developed software solution has been tested by application to many different CAD models. For this purpose, 3D CAD models of industrial as well as experimental sheet metal forming tools designed in CATIA V5 were used (Fig. 10).

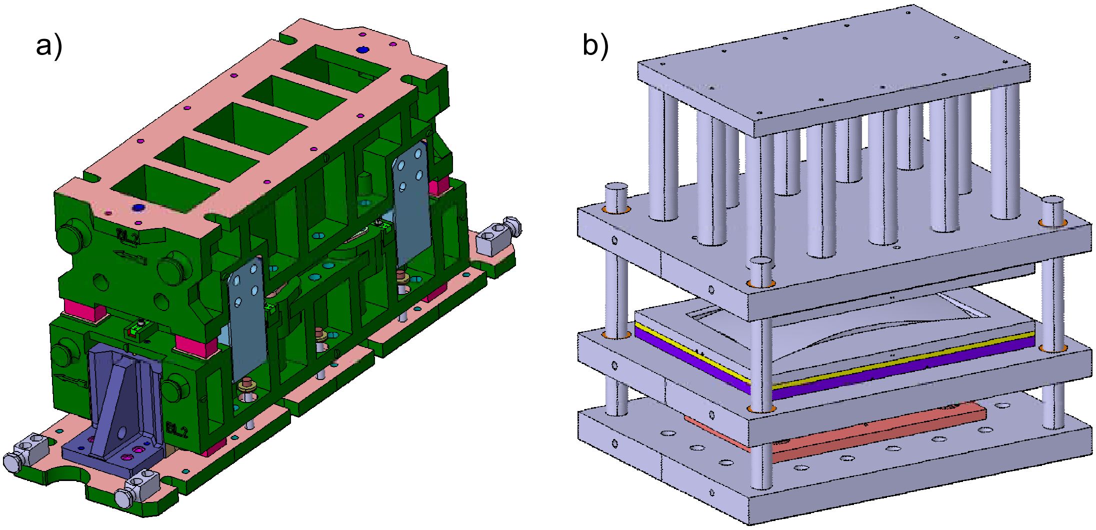

Fig. 10: Investigated industrial a) and experimental b) sheet metal forming tools.

Generally, the CAD models of sheet metal forming tools have many components (e.g. features, geometry, parameters etc.) as well as a large network of dependencies (e.g. relations among parameters or geometry), because of parametric-associative design of these models. For example, the tool model for production of a car bonnet (Fig. 10b) has 9407 components: 603 structural elements, 1834 geometric elements, 5635 parameters, 1335 dependencies and no further external links. It consists of 17 parts and 4 assemblies. In contrast, the model of industrial sheet metal forming tool (Fig. 10a) is organized in 38 parts and 31 assemblies and the number of components and dependencies is nearly four times higher. It should be noted that the model of the industrial tool uses external links to reuse available models and defines interfaces between separate parts. This technique is common in the design of complex CAD models. So managing the complexity of large CAD models is only possible by an interactive approach of the visualization technique.

It has been proven that the use of the CATIA automation interface is generally well suited, but some limitations have to be considered, e.g. external links cannot be traced. Furthermore, it has been shown that the automation interface has a relatively poor performance and needs a long time for the analysis of large CAD models. The performance could be improved by using a more powerful interface (e.g. CAA). A direct integration of the developed visualization in a CAD system would create additional benefits for a design engineer.

8 CONCLUSIONS

In this paper, a new method and software solution for documentation and visualization of complex networks of parametric-associative information within 3D CAD models was presented. For this purpose, the architecture of the developed visualization system (VIPA) was introduced. Based on the general structural analysis of the parametric 3D models, a CAD system-independent graphical language with two forms of visualization was implemented within the software prototype. The structure and parametric-associative information of the model were documented through the XML file. Further, a case study for analysis of sheet metal forming tools was presented.

The methods described in this paper are universal and can therefore not only be used to visualize parametric information in 3D CAD models of complex sheet metal forming tools but can also be transferred to similar applications, e.g. the construction of complex metal forming machines. Implementing the solution for other CAD systems is generally possible.

The presented method will gain importance for the documentation of tool design, especially when combined with a thorough knowledge management. Consistent application of these methods will result in a continuous quality improvement. The innovative visualization concepts can simplify the parametric relations of a complex CAD models considerably.

Additional research could be extended beyond the pure visualization of parametric 3D CAD models. Designers are usually interested in a two-way information exchange to modify the CAD model upon finding failures. For example, the visualization tool could implement the option of inserting, deleting and updating parameters and their associations. These interactions should then be synchronized with the original CAD model. These features could be very helpful for the designer. Another interesting approach is the development of tools for the graphical design of parametric relations from scratch. For those investigations the methods of systems engineering, e.g. SysML, could be used [19].

9 ACKNOWLEDGEMENTS

The authors would like to thank the German Federation of Industrial Research Associations (AiF) for the financial support of this project (AiF-No. 15309 BG) funded by the budget of the Federal Minister of Economics and Technology.

REFERENCES

[1] Larsen, S.; Jensen, C. G.: Converting Topology Optimization Results into Parametric CAD Models, Computer-Aided Design & Applications, 6(3), 2009, 407-418.

[2] Hoffmann, M.C.: Constraint-Based CAD, http://www.13thmonkey.org/documentation/CAD/ConstrBasedCAD.pdf, 03.10.2010

[3] Baxter, D.; Gao, J.; Case, K.; Harding J.; Young B.; Cochrane, S.; Dani, S.: An engineering design knowledge reuse methodology using process modelling, Springer-Verlag London Limited, 2007.

[4] Mohammed, J.; May, J.; Alavi, A.: Application of Computer Aided Design (CAD) In Knowledge Based Engineering, Proceedings of The 2008 IAJC-IJME International Conference, 2008.

[5] Myung, S.; Han, S.: Knowledge-based parametric design of mechanical products based on configuration design method, Expert Systems with Applications 21, Elsevier Science Ltd., 99-107, 2001.

[6] Rappoport, A.: An Architecture for Universal CAD Data Exchange, SM'03, Seattle, Washington, USA, June 16–20, 2003.

[7] Choi, G.-H.; Mun, D.; Han, S.: Exchange of CAD Part Models Based on the Macro-Parametric Approach, International Journal of CAD/CAM, 2(1), 2002.

[8] Spitz, S.; Rappoport, A.: Integrated Feature-Based and Geometric CAD Data Exchange, ACM Symposium on Solid Modeling and Applications, 2004.

[9] Kim, J.; Pratt, M. J.; Iyer, R. G.; Sriram, R. D.: Standardized data exchange of CAD models with design intent, Computer-Aided Design, 40(7), Current State and Future of Product Data Technologies (PDT), 760-777, 2008.

[10] Pratt, M. J.: Exchanging history-based parametric CAD models using ISO 10303, Int. J. Product Lifecycle Management, 4(4), 2010.

[11] Pahl, G.; Beitz, W.: Engineering Design - A systematic Approach, Springer-Verlag, London, 1988.

[12] Prieur, M.: Functional Elements and Engineering Template-based Product Development Process. Application for the Supporting of Stamping Tool Design. PhD thesis, Band 1, Karlsruhe, 2006.

[13] Berglund, L. C.; Jensen, C. G.: Robust Parameterization Schema for CAx Master Models, Computer-Aided Design and Applications, 5(5), 2008, 715-729.

[14] Katz, C. N.: Parametric modeling in AutoCAD, AECbytes, 2007.

[15] Dierneder, S.: Parametric Mechatronic Design - A driver for modern optimised systems and plants, The 2nd International Multi-Conference on Engineering and Technological Innovation 2009 (IMETI 2009), Orlando, Florida, USA, 2009.

[16] Ba ak, H.; Gülesin, M.: A Feature Based Parametric Design Program and Expert System for Design, Mathematical and Computational Applications, 9(3), 2004, 359-370.

[17] Ma, Y.-S.; Tong, T.: Associative feature modeling for concurrent engineering integration, Computers in Industry 51, 2003, 51-71.

[18] Horváth, L.; Rudas, I. J.; Hancke, G.: New Content behind the Concept Intelligent Engineering, Intelligent Engineering Systems (INES), 14th International Conference, 2010.

[19] Wölkl, S.; Shea, K.: A Computational Product for Conceptual Design Using SysML, Proceedings of the ASME 2009 International Design Engineering Technical Conferences & Computers and Information in Engineering Conference (IDETC/CIE 2009), DETC2009-87239, San Diego, California, USA, 2009.

[20] Peak, R. S.; Burkhart, R. M.; Friedenthal, S. A.; Wilson, M. W.; Bajaj, M.; Kim, I.: Simulation-Based Design Using SysML Part 1: A Parametrics Primer, INCOSE Intl. Symposium, San Diego, 2007.

[21] Peak, R. S.; Burkhart, R. M.; Friedenthal, S. A.; Wilson, M. W.; Bajaj, M.; Kim, I.: Simulation-Based Design Using SysML Part 1: Celebrating Diversity by Example, INCOSE Intl. Symposium, San Diego, 2007.

[22] Yang, J.; Han, S.; Cho, J.; Kim, B.; Lee, H.: An XML-Based Macro Data Representation for a Parametric CAD Model Exchange, Computer-Aided Design and Applications, 1(1-4), 2004, 153–162.

[23] Wrobel, G.; Ebert, R.-E.; Pleßow, M.: Graph-Based Engineering Systems - A Family of Software Applications and their Underlying Framework. In: Electronic Communications of the EASST, Vol. 6, Hrsg.: European Association of Software Science and Technology e.V., Berlin, 2007.