Automatic Assembly Design for Engineering-to-Order Products Based on Multiple Models and Assembly Features

Iraklis Chatziparasidis, Nickolas S. Sapidis

Automatic Assembly Design for Engineering-To-Order Products based on Multiple Models and Assembly Features

Iraklis Chatziparasidis [1,2], Nickolas S. Sapidis [2],

[1] Research & Development Department, Kleemann Hellas SA, Kilkis Industrial Area 61100, P.O. Box 25, Greece, http://www.kleemannlifts.com/ ihatz@kleemann.gr

[2] Department of Mechanical Engineering, University of Western Macedonia, Kozani 50132, Greece http://mech.uowm.gr/index.php/staff/317 nsapidis@uowm.gr

Abstract. When it comes to Engineering-To-Order (ETO) products, neither the exact number nor the form of the components in them can be predefined. Thus, existing assembly models and generative design techniques are not adequate to support development of design automation tools for ETO products. ETO companies usually use custom libraries with past case designs that are adjusted to a customer's requirements. This method is not cost effective and it is prone to human errors. In this work, we present the *Automatic Assembly Synthesis Model* (*AASM*), connecting a Knowledge Based Engineering (KBE) system and a CAD system to automate routine design tasks for ETO mechanical products.

Keywords: Automatic Synthesis; Assembly Features; Assembly Model.

1 Introduction

Today, in the context of a globalized market, customers have high demands for products that are tailored to their individual needs and are offered at a price that is very close to that of mass-produced products. Engineering-To-Order (ETO) companies offer to their clients products that are tailored to their needs. These companies are forced to reduce costs and lead time to gain an advantage over the competition. A major cost factor for ETO companies is the time required for the product to be designed and engineered, and for manufacturing drawings to be published and launched to the shop floor. In most cases, these companies have a number of premade 3D models (and the corresponding manufacturing drawings) and modify them to adjust the dimensions, the function and/or the aesthetics of the product to the customer requirements. Unfortunately, this method is prone to human errors, and these errors may create extra remanufacturing costs. An ETO company would gain a significant advantage by using a tool that would create automatically 3D assembly models for its products. Today, CAD systems can be used to partially automate some routine design operations when these are combined with generative modeling methodologies [1,2]. However, this kind of tools can be used only in situations where all component configurations in the final

product are known. Often this is not true, as many ETO companies prefer more flexible approaches to product configuration, employing, e.g., Knowledge Based Engineering (KBE) systems or Rule Based Management Systems (RBMS). These systems can be employed to effectively capture knowledge by storing technical guidelines, relations, facts [3], "best practices", and even a company's commercial and business rules. However, on the side of CAD systems a major limitation remains: it is extremely difficult to have pre-defined 3D assembly models matching all possible product configurations implied by a KBE system. Thus, this research proposes development of a routine-design automation tool that will create any 3D assembly model from scratch by synthesizing appropriate components. Since existing assembly modeling methods are not well suited for design automation procedures, we present, in this work, the *Automatic Assembly Synthesis Model (AASM)*, a method to link a KBE system and a CAD system.

2 Literature Review

2.1 Assembly Models & Assembly Features

According to Demoly et al [4], Assembly Modeling deals with the definition of an informational product model including all product components and the related relationship information. In [5], we present an extensive review of the related research of the last 15 years, making clear that current assembly models are not adequate for automating assembly procedures. Existing assembly models, like the OAM [6], the AREP [7], the FGT [8], or those discussed in [9,10], do not present a method to support transfer of product configuration information, from a KBE or RBS system to a CAD system, in a way that would facilitate automatic assembly synthesis. Thus, we focus, in this paper, on exactly presenting an assembly model and methodology appropriate for automatic assembly synthesis for ETO products.

In feature-based product modeling, the Assembly Feature (AF) is an important concept describing relationships and interaction regions between parts, however, currently there is no unified definition for it [11]. In [9], an AF is defined as an information carrier for assembly-specific information. In [8], an AF is defined as a pair of geometry features restricted by a specific assembly constraint. According to [12], an AF represents a region of a component that is of interest in the assembly context. In [6], an AF specifies relationships in a pair of assembled components. In [13], an AF is defined as an association between two form features which are on different parts. In [14], an AF is defined as a generic "solution" referring to two groups of parts that need to be related by a relationship to solve a design problem. In [15], mating features are defined as those features that comprise mating relations between parts to be assembled. In [16,17], a method that simplifies complex products to achieve a virtual assembly modeling process in real time is presented. Here, form features are defined as generic shapes useful in computer-aided design applications, and assembly features are the connections between form features. In [7], an AF is defined as a property of an Assembly Unit (AU) providing assembly related information. An AU can be a sub-

assembly, a component or an envelope. Envelopes are volumes within which parts and sub-assemblies are to be designed. In [18], the following concepts are presented: a) Design Spaces, which are simplified objects defining the area that will be occupied by each component when the detail design phase will be completed. b) Constraints, which describe, in an algebraic manner, the kinematic relationships between design spaces. c) Interface Features, which are geometric entities that are used as connection interfaces between Constraints and Design Spaces. d) Layout Components: These are produced by combinations of Design Spaces with Interface Features. e) Connection Features: These are detailed form features that are designed by the designer and aim to implement the corresponding Interface Features. This methodology has as a priority to ensure the kinematic functionality of the assembly before proceeding to the detailed geometric design of individual components. Xu et al. [11] present the concept of Interaction Features Pair (IFP) describing how components interact with each other at the assembly creation stage. In [19], the authors propose a Feature-based design method, that focuses on modeling complex relations among features. Four kinds of features are proposed: a) Conceptual Features, b) Assembly Features (AFs), c) Component Basic Features, and d) Component Detail Features. Ma et al. [20–22] introduce the concept of Associative Features which are features that cannot be represented using conventional features. An example is the cooling channels in a mold, which are represented as CAD solids called "cooling solids". Thus, cooling channels are easily created by applying the solid-modeling subtract operator on the cooling solids and the initial mold. Dixon [23] presents a system that automatically identifies AFs. First, the user teaches the system interactively by examples of AFs. These AFs are then used as "standards" by the system to identify AFs in assembly models that are saved in a neutral format (e.g., STEP). Kim et al., [24] introduce a formalism and associated tools to capture joining relations in assemblies. In this work, a Mating Feature is defined as a set of component geometric-entities used to assemble parts. In [25], the use of assembly features in standard parts (e.g. bolts, nuts etc) is proposed. In [26], Connection Features are functional relationships representing the internal degrees of freedom that the corresponding form features must have, to implement a specific connection type. In [27], a system for supporting rapid assembly modeling of standard parts is presented. The system is based on the concept of Typical Assembly Feature (TAF), defined as a geometric element of a component which can constrain and orient this component in an assembly. Singh and Bettig [28] have presented the concept of "assembly ports" as a method to embed assembly information into the part model in order to automate the process of applying mating constraints. An assembly port is defined to be a group of one or more low-level geometric entities, such as faces, edges, or centerlines, that undergo mating constraints in order to join parts in a CAD assembly. In [29,30], the authors propose a framework method to integrate assembly modeling and simulation based on Assembly Feature Pairs. An Assembly Feature Pair consists of form-feature pairs containing information on assembly behaviors.

2.2 "Half" Assembly Constraints

Most contemporary CAD systems have tools implementing the concept of "half" assembly constraints, e.g., in *PTC Creo* they are called *Component Interfaces* [31], in

Autodesk Inventor they are called *iMates* [32], etc. These tools allow the designer to store assembly constraints, in advance, in each component, during the design phase. These tools aim to cut down the time a user spends to assemble components. However, mere use of these tools alone cannot fully automate assembly synthesis, because of the lack of any information about the requested assembly's structure. In the present *AASM* model the concept of "half" constraints has been integrated as a fundamental block into the concept of *Assembly Feature (AF)*. By integrating the concept of "half" constraints, in the form of the *Semi-Constraint* object, into the AASM model, we provide a framework that can fully automate the assembly synthesis procedure. This way, 3D part models, that have not been designed together in an assembly, can be automatically connected if they contain compatible AFs.

3 The Automatic Assembly Synthesis Model (AASM)

The design automation procedure, proposed here, is based on the use of "generative part models" which generate the part instances that compose the desired 3D assembly. A generative part model differs from a single geometric part-model. While a geometric part model has fixed dimensions and features, the generative part model is a generic representation of the part and it is used to create instances with varying form and dimensions. Generative part models also contain special form features that are used as connection ports, named *Assembly Features* (*AFs*).

The *Automatic Assembly Synthesis Model (AASM)* includes two major components: The *Schematic Assembly Model (SAM)* and the *Intermediate Assembly Model (IAM)*; see Fig. 1. The *Schematic Assembly Model* is a preliminary model that converts the structural rules, that are stored within a KBE system (e.g., in an IF...THEN...ELSE form), into an object-oriented assembly-structure form that functions as a configuration rule guiding the automatic assembly synthesis procedure. The SAM contains information on the structure of the desired 3D assembly and the connection types that must be applied on corresponding components. The SAM does not contain detailed information regarding how these connections will be implemented at the 3D geometry level in the CAD system. The IAM is an augmented implementation of the SAM. The IAM is based on the SAM regarding assembly structure information but it does also contain detailed information specifying which *Assembly Features* of each component must be used for the 3D assembly to be created.
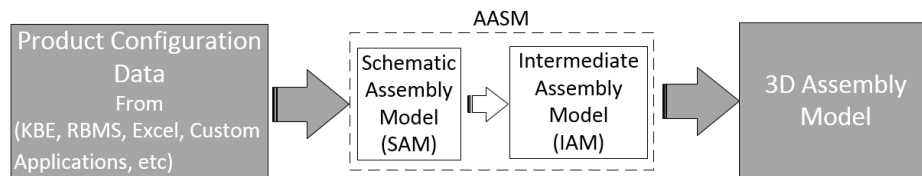

Fig. 1. AASM Model

In short, the SAM describes the assembly that the KBE system requires and the IAM represents the corresponding 3D assembly model that the CAD system will

create. The separation between initial description and final implementation is one of the major attributes of AASM making it adequate for automatic synthesis of complex assemblies with significant variation in their configurations. Dividing AASM into two sub-models (SAM and IAM) results also into an increased flexibility, when it comes to implementation of a Design Automation tool for large teams of designers.

3.1 Assembly Feature (AF): A New Definition for the Automatic Assembly Synthesis Model

In this work, *Assembly Feature* is a graphical formation of the 3D component model that functions as a connection port allowing parts with compatible *Assembly Features* to be automatically connected. AFs are created by the designer during the design of each generative part model and are represented in an object-oriented manner within the AASM. An *Assembly Feature* is composed of graphical entities, which can be either B-Rep entities (like: vertices, edges or faces) or datum graphical objects (like: points, axes or planes) or a combination of these. These graphical entities will be matched, using specific assembly constraints, with the corresponding entities of the associated AF. Matching of these entities is achieved through embedded information in the form of attributes within the B-Rep model. Each of the entities forming an AF also includes a number of *Assembly Feature Attribute Pairs (AFAP)*; see Fig. 2. Each AFAP contains a reference to its parent graphical entity and to the type of a *Semi-Constraint* that must be used. Entities with compatible AFAPs are automatically matched via a matching algorithm. Compatible AFAPs are considered these that contain references to graphical entities of the same type and identical Semi-Constraint types. A *Semi-Constraint* is a special type of assembly constraint. *Semi-Constraints* are a way to define assembly constraints and assign them to the part model before assembling it into an assembly. A *Semi-Constraint* is the "half" of an assembly constraint and is added to each of the corresponding components independently. Two *Semi-Constraints* have to be combined to form a complete assembly constraint. A *Semi-Constraint* is a label stored as attribute within the corresponding part of the B-Rep model. It is the CAD system that implements these labels by changing the position and orientation of the components. In most contemporary CAD systems, each part model and each assembly model has its own coordinate system. The position of a component within an assembly is specified by matching the position and the orientation of the component's coordinate system relatively to the coordinate system of the assembly. The definition and/or the change of position of a component within an assembly are controlled by *Transformation Matrices*. Every movement or rotation of a part is translated into transformation of these matrices so that they define the new position and orientation of the part.
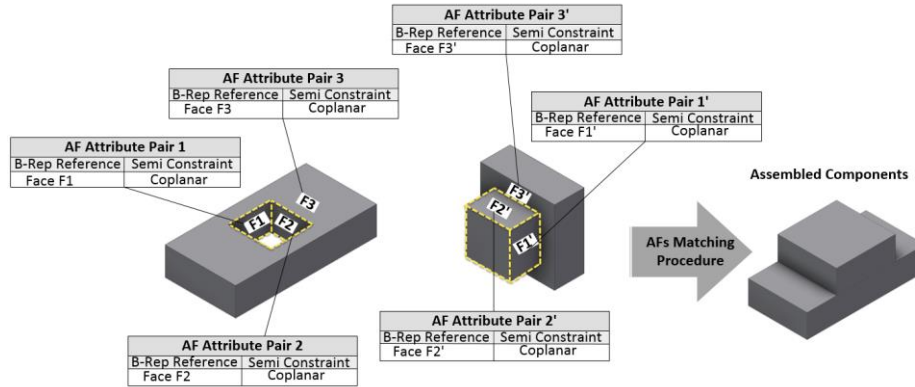
Fig. 2. Assembly Features and Assembly Features Attribute Pairs

Three are the types of AFs used here: *Form Assembly Features (FAFs)*, *Skeleton Assembly Features (SAFs) and Composite Assembly Features (CAFs)*. *Form Assembly Features (FAF)* are these AFs that totally coincide with a corresponding form feature. *Skeleton Assembly Features (SAF)* are *Assembly Features* formed by auxiliary geometric entities like *Planes*, *Axes and Points*. SAFs can be used to represent a connection between parts when adequate FAFs are not present. *Composite Assembly Features* are used when the corresponding form feature that will be used as connection port does not provide all the necessary B-Rep entities to implement the connection. In these situations, auxiliary entities are used to complete the geometric description of the connection. The scope of most AFs is limited only at part level, meaning that these AFs are used only to connect the related part to other parts. However, there are cases where an AF that belongs to a specific component must function also as *Assembly Feature* of a newly-formed sub-assembly, to allow this sub-assembly to be connected with other components, forming another assembly. For these reason, all types of AFs can be declared to be *External Assembly Features*.

3.2 AASM: The Schematic Assembly Model (SAM)

The SAM consists of the *SAM Structure* object and the *SAM Connection Rules Collection* object (Fig. 3). The *SAM Structure* is a tree-based hierarchical structure object, resulting from the KBE system. The final 3D assembly model, that will be automatically synthesized, has to comply with the *SAM Structure*. This is the substantial difference between the *SAM Structure* object, presented in this work, and the previous approaches in the literature where tree-based hierarchical structures are used to describe the structure of an assembly after this is constructed by the CAD user [33–36]. The *SAM Structure* contains only information on what components are included in the assembly. Information about the connection relations between components is stored in the *SAM Connection Rules Collection*. Each row of the *SAM Connection Rules Collection* (Fig. 3) represents a *SAM Connection Rule*. The "*:*" symbol is used to represent the relation between the related objects. The *SAM*

Connection Rule is an object-oriented representation of the relationship between two components linked together with a kinematic relationship called "*Kinematic Pair*". A component can be either a part or an assembly. The *SAM Component* is the base class for the *SAM Part* and *SAM Assembly* objects that derive from it. The *SAM Part* object represents a component that cannot be decomposed into components and the *SAM Assembly* object represents an assembly or a sub-assembly. Each *SAM Component* child instance object has member variables, numerical or boolean, named as *Attributes*, that control the form and the dimensions of the corresponding generative part model instance.

Chen et al [33,37] propose six kinematic pair types: *Prismatic Pair*, *Revolute Pair*, *Screw Pair*, *Cylindrical Pair*, *Spherical Pair* and *Planar Pair*. Demoly et al [4] propose ten kinematic pairs: *Rigid, Revolute, Prismatic, Screw, Cylindrical, Spherical, Planar, Point-contact, Line-contact* and *Curve-contact*. Finally Csabai et al [18] propose fourteen kinematic relations: *Distance, Spherical, In-plane, In-line, On-cylinder, Mate, Align, Cylindrical, Co-directional, Revolute, Prismatic, Universal, Screw* and *Rigid*. In this work, the role of the *SAM Kinematic Pairs* differs significantly from previous approaches. A *SAM Kinematic Pair* provides a description of the relative motion existing between two components. A *SAM Kinematic Pair* does not contain information on how this kinematic relationship can be implemented in the 3D assembly model. This kind of information is provided by the IAM, which will de described in Section 3.3. In this work, the AASM uses seven *SAM Kinematic Pairs*: *Rigid, Prismatic, Spherical, Cylindrical, Contact, Angular* and *Insert* (Table 1). The *Rigid, Prismatic, Spherical* and *Cylindrical* kinematic pairs are adopted from [4,18,33]. The *Contact* kinematic pair is added instead of a *Planar* since it can better describe the contact between planar and cylindrical faces. The *Insert* kinematic pair is added because it can better describe bolted connections and bearing-shaft type relations. Finally, the *Angular* kinematic pair is added to represent very common situations of angular relationships in mechanical assemblies like hinge-type connections.
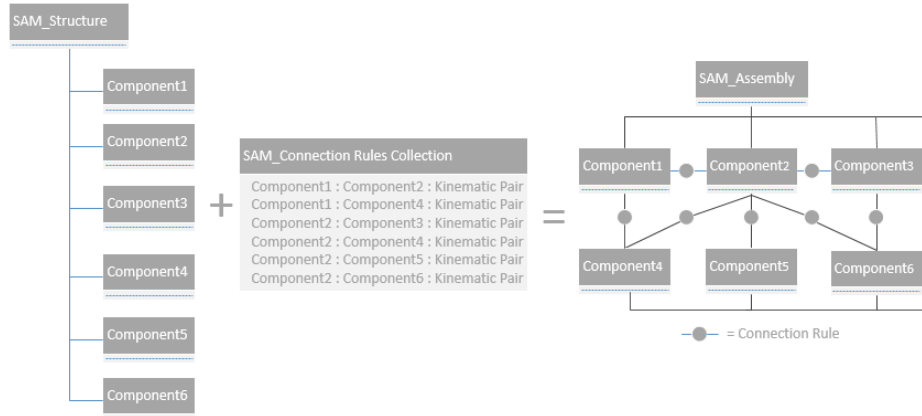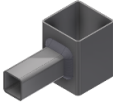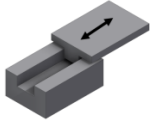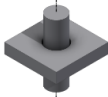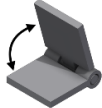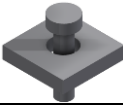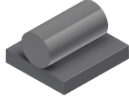

Fig. 3. The Schematic Assembly Model consists of the SAM_Structure and the SAM_Connection Rules Collection

Table 1 - Kinematic Pairs

Kinematic Pair	Description
Rigid	Two components cannot be moved relatively to each other. These situations occur for example if these components are welded, bolt connected, pressed to fit together or other components prevent them to move.
Prismatic	One component can slide relatively to another.
Cylindrical	A cylindrical component (e.g. a shaft) is placed coaxially on cylindrical features (e.g. sliding bearings) of another component.
Angular	"Hinge" type connection between two components.
Insert	One component is inserted into hole/socket features of the second component e.g. a bolt inserted into a hole.
Spherical	Two components share a virtually common centre.
Contact	Two components are in contact at a line. This connection type is usually combined with other connection types. It could be used to indicate contact between two planar surfaces, or between two cylindrical surfaces, or between a planar and a cylindrical surface.

By dividing the SAM into two parts (*SAM Structure* and *SAM Connection Rules Collection*) the model is better suited for cases where the same components with different connection rules can produce different valid assemblies.

3.3   AASM: The Intermediate Assembly Model (IAM)

The *Intermediate Assembly Model (IAM)* fills the informational gap between SAM and the implemented 3D-CAD assembly model. IAM contains all the information on how components can be connected. IAM is created in four steps:

During the first step, the initial structure of the IAM, based on the SAM, is created. For each *SAM Component* a corresponding *IAM Component* is created, and for each *SAM Connection Rule* an *IAM Components Association* object is created. An *IAM Component* is the base object for the *IAM Assembly* and *IAM Part* objects which are derived from it. An *IAM Assembly* object represents either a 3D sub-assembly that is part of a larger assembly (sub-assembly) or the final 3D assembly. An *IAM Part* object represents a component that cannot be further decomposed into components. Each instance of the *IAM Part* class includes a member function that is connected with the corresponding generative part model. During the second step, this member function generates all the corresponding 3D part model instances, but the IAM does not yet contain the information on how these part models should be connected to form a 3D assembly. This information is obtained during the third step (Fig. 4) where the *IAM Component Associations* objects, created at the first step, are completed with information on the AFs that will be used to form the corresponding connections in the 3D assembly.
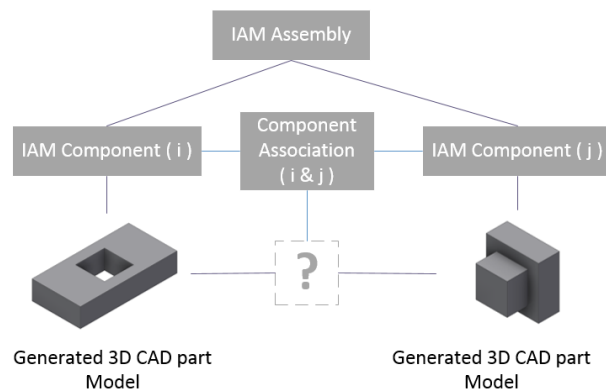

Fig. 4. Step 2: Generation of 3D Part Models

During the third step, all 3D part models, created at the second step, are scanned and pairs of compatible AFs are associated through an *Assembly Feature Association object (AFA)*. For each compatible AF pair found, a new AFA object is created. The created AFAs are then associated with the corresponding *IAM Component Association* objects. An AFA object contains references to each of the associated *Assembly Features* and to a number of *Assembly Feature Attribute Pair Associations (AFAPA)*. An AFAPA is formed by two matched *Assembly Feature Attribute Pairs (AFAP)* (Fig. 5). This step completes construction of IAM (Fig. 6).

During the fourth step, *Component Association* objects (created in Step 1) are checked for compatibility to the corresponding *SAM Connection Rules*. Compatibility between a *SAM Connection Rule* and a *Component Association* object is confirmed through the compatibility between the *SAM Kinematic Pairs* and the *Assembly Feature Attribute Pairs* that implement them.
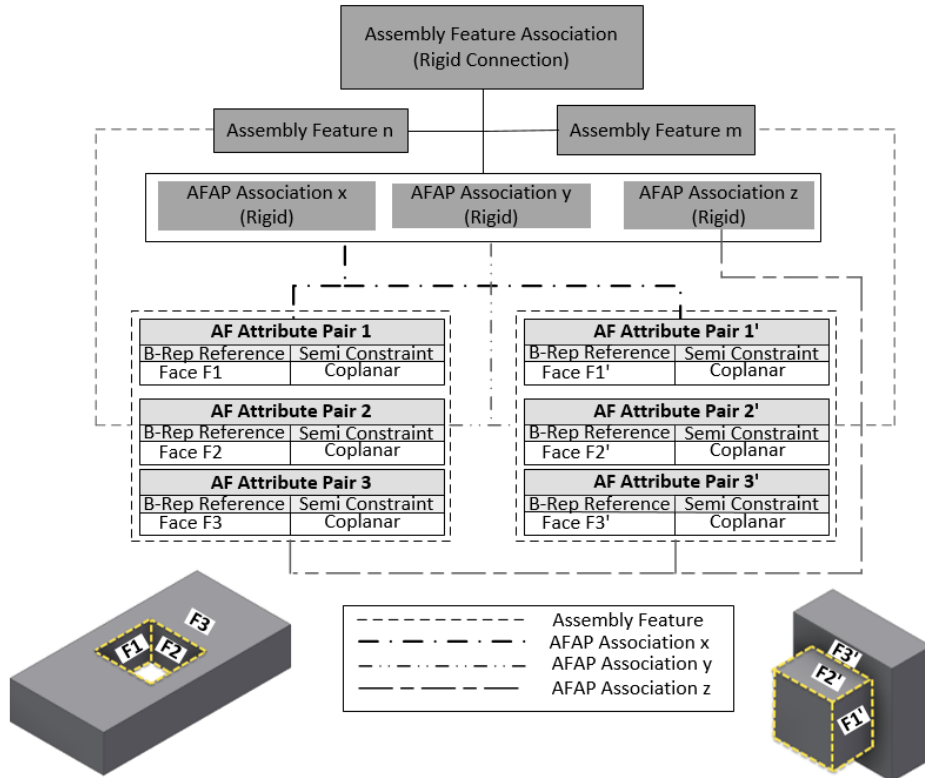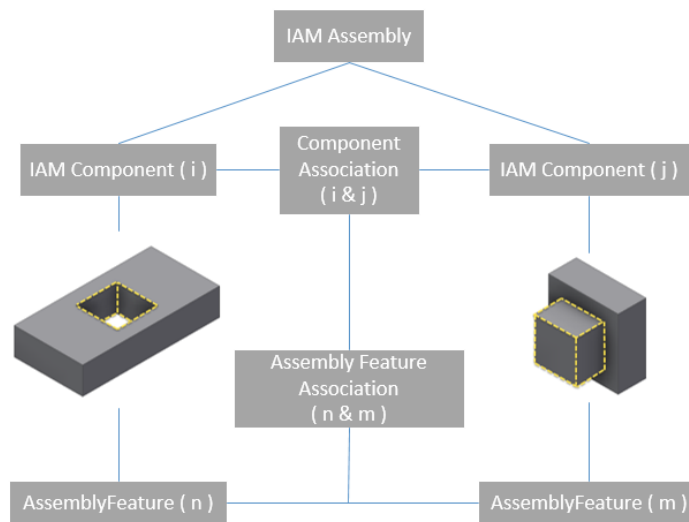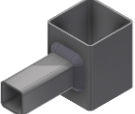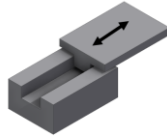
Fig. 5. AF Association
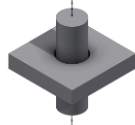
Fig. 6. Step 3: AF Association

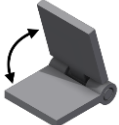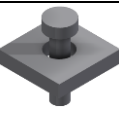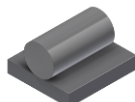In Table 2, we present *SAM Kinematic Pairs* and AFAPs that implement them. Besides compatibility checks, connectability checks should also take place [28]. Connectability refers to the ability of two parts to become connected without the occurrence of "solid-solid interference". For connectabilty checks, an algorithm, based on CAD tools for interference and collision detection, is used.

Table 2 - Kinematic Pairs and implementations using AFAPs

Kinematic Pair	Assembly Feature Attribute Pairs Implementation
Rigid	Any combination of *AFAP*s that remove all the degrees of freedom between two components, e.g: • three "Plane : Coplanar" AFAPs. • or two "Plane : Coplanar" plus one "Plane : Align" AFAPs. • or one "Plane : Coplanar" plus two "Plane : Align" AFAPs. • or three "Edge : Align" AFAPs , etc.
Prismatic	Any combination of *AFAP*s that allows only one-directional linear movement, e.g.: • two "Plane : Coplanar" AFAPs. • or two " Edge : Align" AFAPs. • or a combination of the above two.
Cylindrical	An "Axis : Align" *AFAP*.
Angular	A combination of: • one "Axis : Align" AFAP plus one "Surface : Angle" AFAP. • or one "Edge : Align" AFAP plus one "Surface : Angle" AFAP.
Insert	Any combination of *AFAP*s that leaves only one rotational degree of freedom remaining, e.g.: • one "Plane : Coplanar" and one "Axis : Align" AFAPs. • or one "Plane : Align" and one "Axis : Align" AFAPs.
Spherical	All linear movement degrees of freedom are removed, all rotational degrees of freedom remaining: • one "Point : Coincidence" AFAP.
Contact	A "Surface : Tangent" *AFAP*.

4   Implementation of the Automatic Assembly Synthesis Model

To test the effectiveness of the AASM, this has been implemented in a software system automating the synthesis of elevator-car 3D assembly models. This system includes four major components: a) a commercial Rule Based System (IBM's ILOG), b) a commercial database system (Microsoft SQL Server), c) a commercial 3D CAD system (Autodesk Inventor) and d) the present Automatic Assembly Synthesis Model (AASM), developed as a CAD add-in, named as *CabinsKBE*, materializing this design automation workflow: [a] The customer's order is passed as input to the Rule Based System (RBS). [b] The RBS produces a detailed description for the elevator car, which is stored in the database. [c] The CAD system, extended with the *CabinsKBE* add-in, retrieves the stored configuration from the database, creates the corresponding SAM and IAM, generates all required 3D parts and, finally, synthesizes them into the 3D assembly.

CabinsKBE has managed to successfully assemble various types of elevator cars. Fig. 7A presents an example of a panoramic-car assembly model synthesized by the present software. For this model, *CabinsKBE* created: 30 *SAM* and *IAM Component* instances, 69 *SAM Connection Rules* and *IAM Components* Associations, and 138 *AF*s. Fig. 7B presents an example of a goods passenger car. For this assembly the AASM software created: 20 *SAM* and *IAM Component* instances, 23 *SAM Connection Rules* and *IAM Components* Associations, and 42 AFs. In both cases, all components were generated and then synthesized, into the desired assembly, by *CabinsKBE* in a fully automatic mode.
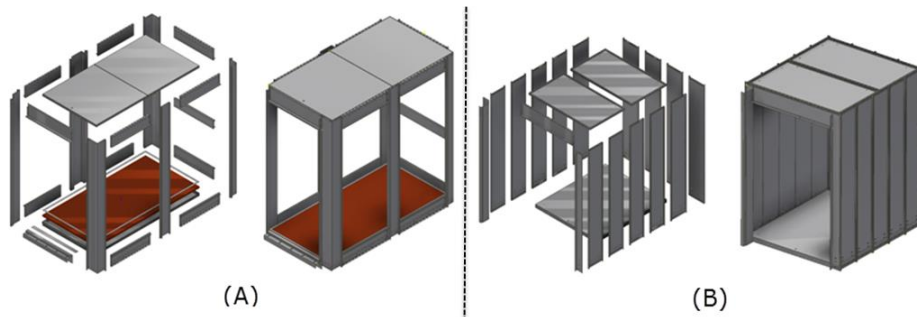
Fig. 7. A Panoramic Car (A) and a Passenger Elevator Car (B) Automatically Synthesized using CabinsKBE.

5   Discussion - Conclusions

When it comes to Engineering-To-Order (ETO) products (e.g., products for which neither the number nor the form of components can be standardized), existing CAD tools and automation methodologies are not capable to support full automation of routine design procedures. The *Automatic Assembly Synthesis Model (AASM),* described above, has been shown adequate to: fill the communication gap between KBE (and/or Rules Based Systems [RBS]) and CAD systems, support automatic

assembly synthesis, and construct valid 3D assembly models. The substantial features of AASM are its dual structure and the use of *Assembly Features* as they are redefined here. The dual structure of AASM makes it adequate to represent (a) the configuration structure implied by a KBE or a RBS system and (b) the specific assembly implementing this configuration. The effectiveness of the AASM model has been tested in a design automation system developed for elevator cars. The implemented system, at Kleemann Hellas SA, decreased significantly the time spent to process each car order. For example, the design time for a panoramic elevator car was reduced from six hours, when generative techniques and pre-designed assembly models are used, to fifteen minutes with the implemented AASM (*CabinsKBE*) software. Other benefits were that costs caused by human errors were eliminated and lead times for product delivery were significantly reduced.

References

1. Krish S: A practical generative design method. CAD Computer Aided Design, vol. 43, pp. 88–100 (2011).
2. Skarka W: Application of MOKA methodology in generative model creation using CATIA. Engineering Applications of Artificial Intelligence, vol. 20, pp. 677–690 (2007).
3. Amadori K, Tarkian M, Ölvander J, Krus P: Flexible and robust CAD models for design automation. Advanced Engineering Informatics, vol. 26, pp. 180–195 (2012).
4. Demoly F, Toussaint L, Eynard B, Kiritsis D, Gomes S: Geometric skeleton computation enabling concurrent product engineering and assembly sequence planning. Computer-Aided Design, vol. 43, pp. 1654–1673 (2011).
5. Chatziparasidis I, Sapidis NS: Automatic Assembly Synthesis based on Multiple Models and Assembly Features (Technical Report, Depart. of Mechanical Engineering, UOWM). (2016).
6. Rachuri S, Han YH, Feng SC, Roy U, Wang F, Sriram R, et al.: Object-oriented representation of electro-mechanical assemblies using UML. NIST IR, vol. 7057, (2003).
7. Shyamsundar N, Gadh R: Internet-based collaborative product design with assembly features and virtual design spaces. Computer-Aided Design, vol. 33, pp. 637–651 (2001).
8. Qi F: A online retrieving method for product functional and structural information based FGT model. WSEAS Transactions on Computers, vol. 8, pp. 1749–1759 (2009).
9. van Holland W, Bronsvoort WF: Assembly features in modeling and planning. Robotics and Computer-Integrated Manufacturing, vol. 16, pp. 277–294 (2000).
10. Brunetti G, Golob B: A feature-based approach towards an integrated product model including conceptual design information. Computer-Aided Design, vol. 32, pp. 877–887 (2000).
11. Xu Z, Zhang J, Li Y, Jiang S, Sun Y: Product modeling framework based on interaction feature pair. Computer-Aided Design, vol. 45, pp. 1591–1603 (2013).
12. Sugimura N: JNC proposal of STEP assembly model for products. (2000).
13. Shah JJ, Rogers MT: Assembly modeling as an extension of feature-based design. Research in Engineering Design, vol. 5, pp. 218–237 (1993).

14. Deneux D: Introduction to assembly features: An illustrated synthesis methodology. Journal of Intelligent Manufacturing, vol. 10, pp. 29–39 (1999).
15. Chang C-F, Perng D-B: Assembly-part automatic positioning using high-level entities of mating features. Computer Integrated Manufacturing Systems, vol. 10, pp. 205–215 (1997).
16. Xiao H, Cheng H, YU J, Li Y: Dynamic assembly simplification for virtual assembly process of complex product. Assembly Automation, vol. 34, pp. 1–15 (2014).
17. Yu JF, Xiao H, Zhang J, Cheng H, Xin B: CAD model simplification for assembly field. International Journal of Advanced Manufacturing Technology, pp. 1–13 (2013).
18. Csabai A, Stroud I, Xirouchakis PC: Container spaces and functional features for top-down 3D layout design. CAD Computer Aided Design, vol. 34, pp. 1011–1035 (2002).
19. Yin CG, Ma YS: Parametric feature constraint modeling and mapping in product development. Advanced Engineering Informatics, vol. 26, pp. 539–552 (2012).
20. Ma YS, Britton GA, Tor SB, Jin LY: Associative assembly design features: Concept, implementation and application. International Journal of Advanced Manufacturing Technology, vol. 32, pp. 434–444 (2007).
21. Ma YS, Tong T: Associative feature modeling for concurrent engineering integration. Computers in Industry, vol. 51, pp. 51–71 (2003).
22. Ma Y, Britton GA, Tor SB, Jin L, Chen G, Tang S, et al.: Design of a feature-object-based mechanical assembly library. Computer-Aided Design and Applications, vol. 1, pp. 397–403 (2004).
23. Dixon A, Shah JJ: Assembly feature tutor and recognition algorithms based on mating face Pairs. Computer-Aided Design and Applications, vol. 7, pp. 319–333 (2010).
24. Kim K-Y, Wang Y, Muogboh OS, Nnaji BO: Design formalism for collaborative assembly design. CAD Computer Aided Design, vol. 36, pp. 849–871 (2004).
25. Du B, Wang X, Feng Y, Yu D, Xu G: Intelligent Assembly Technology Based on Standard Parts Feature of CATIA. Modern Applied Science, vol. 8, pp. 49–55 (2014).
26. Noort A, Hoek GFM, Bronsvoort WF: Integrating part and assembly modelling. Computer-Aided Design, vol. 34, pp. 899–912 (2002).
27. Li GD, Zhou LS, An LL, Ji JF, Tan CB, Wang ZG: A system for supporting rapid assembly modeling of mechanical products via components with typical assembly features. International Journal of Advanced Manufacturing Technology, vol. 46, pp. 785–800 (2010).
28. Singh P, Bettig B: Port-compatibility and connectability based assembly design. Journal of Computing and Information Science in Engineering, vol. 4, pp. 197–205 (2004).
29. Zhang J, Xu Z, Li Y, Jiang S: Framework for the integration of assembly modeling and simulation based on assembly feature pair. The International Journal of Advanced Manufacturing Technology, vol. 78, pp. 765–780 (2014).
30. Sun YL, Li Y, Zhang J, Xu ZJ: An Automatic Product Assembly Method Based on Assembly Feature Pair. Advanced Materials Research, vol. 655–657, pp. 1697–1701 (2013).
31. Toogood R: Creo Parametric 3.0 Advanced Tutorial: SDC Publications (2015).
32. Banach D, Jones T, Kalameja A: Autodesk Inventor 2010 Essentials Plus. Cengage Learning (2009).
33. Chen X, Gao S, Yang Y, Zhang S: Multi-level assembly model for top-down design of mechanical products. Computer Aided Design, vol. 44, pp. 1033–1048 (2012).
34. Lee K, Gossard D: A hierarchical data structure for representing assemblies: part 1. Computer-Aided Design, vol. 17, pp. 15–19 (1985).
35. Zeid I: CAD/CAM theory and practice. Vol. 6. New York: McGraw-Hill (1991).
36. Lee K: Principles of CAD/CAM/CAE systems. Addison-Wesley (1999).
37. Chen X, Gao S, Guo S, Bai J: A flexible assembly retrieval approach for model reuse. Computer-Aided Design, vol. 44, pp. 554–574 (2012).