Product Feature Modelling for Integrating Product Design and Assembly Process Planning

Baha Hasan, Jan Wikander

***Abstract*—**This paper describes a part of the integrating work between assembly design and assembly process planning domains (APP). The work is based, in its first stage, on modelling assembly features to support APP. A multi-layer architecture, based on feature-based modelling, is proposed to establish a dynamic and adaptable link between product design using CAD tools and APP. The proposed approach is based on deriving "specific function" features from the "generic" assembly and form features extracted from the CAD tools. A hierarchal structure from "generic" to "specific" and from "high level geometrical entities" to "low level geometrical entities" is proposed in order to integrate geometrical and assembly data extracted from geometrical and assembly modelers to the required processes and resources in APP. The feature concept, feature-based modelling, and feature recognition techniques are reviewed.

***Keywords*—**Assembly feature, assembly process planning, feature, feature-based modelling, form feature, ontology.

I. Introduction

PRODUCT assembly, and of course disassembly, are important aspects of a product's life cycle; as an example, Cho [1] states that almost 53% of product manufacturing time is consumed in carrying out assembly tasks. The integration of product design and APP thus has a strong impact on product realization, as also discussed by Du and Zha [2]. Product features as a concept have been used by several researchers to improve the efficiency of process planning both in manufacturing and assembly [3]-[6]. Features are used when designing a product (Feature-based design), while in process planning features are to be extracted from the product model. The extracted features represent a natural link between product design and process planning domains and provide a valuable mechanism for information exchange [7].

In this paper, features in a product design assembly will be used to bridge the gap between assembly design and APP. APP is the phase that determines the necessary sequence of operations and actions to assemble two or more parts. Since the efficiency of APP is highly dependent on the way in which assembly is modeled, an assembly design model has to be created based on the extracted assembly knowledge that can provide enough information about the relations between connected components/parts and precedence constraints between the connections.

B. Hasan is a PhD student in the machine design department- mechatronics division- KTH University, Stockholm, Sweden (phone: +46760834218, e-mail: bahasan@kth.se).

J. Wikander is a professor in the machine design department- mechatronics division- KTH University, Stockholm, Sweden (phone: +46 8-7907370, e-mail: janwi@kth.se).

The work presented in this paper is a part of a larger effort to develop methods and tools for a more automated and bidirectional link between product CAD and the different processes and resources applied in APP. This work is based, in its first stage, on extracting assembly knowledge from the CAD file by using feature recognition techniques. The second stage is to model the extracted assembly knowledge to support integration with APP. The third and last stage is to share assembly knowledge via a layered ontology structure, which will serve as a communication mean between assembly design and APP.

The presented work is an attempt to give an overview of the three stages of the proposed approach based on an understanding of feature based techniques. The paper is structured as follows: section II reviews related research about features; the two complementary research areas related to the feature concept, namely feature-based modelling and feature recognition. Section III introduces the proposed approach for integrating product design knowledge and APP based on the recognized features. Section IV proposes a layered ontology as a basis for sharing of the recognized product design knowledge. Section V draws a conclusion and provides a summary.

II. Feature Technology

As features spread from product design to different application areas, several definitions for features have been proposed in published literature. One of the "general" definitions for features is that:

> *"a feature is a partial form or a product characteristic that is considered as a unit and that has a semantic meaning in design, process planning, manufacture, cost estimation or other engineering discipline"* [8], where form feature is defined as *"Form features are specific configurations on surfaces, edges, or corners of a part such as holes, slots etc. that carries some engineering meaning"* [9].

More definitions for features and form features are presented in [11].

Features are typically grouped into more specialized classes of the general feature concept. Those classes are characterized by a specific function, which gives the feature its specialized name. For example, "assembly features" are considered as a special type of features formed to capture assembly knowledge related to assembly design, and where these features are used to connect two associative form features of different parts [12].

Assembly features have been used both in design and process planning as well as in design-process integration for mechanical assembly. Since they have been used in different applications, many definitions have been proposed for assembly features from the perspectives of product, process plan, and product-process integration. An example of assembly feature definition, from the product perspective, is the one presented by Chan and Tan [14]: *"assembly features are elementary connection features containing mating relations between the components"*. Another example which describes assembly features from a process plan perspective is the one proposed by Anjum [32]: *"Particular form features that affect assembly operations"*. A third category of definitions, for assembly features, combines both product and process plan perspectives, such as the one presented by Coma et al. [15]:

> *"Any topological, geometrical, technological or functional information assigned to a face, a part or a sub-assembly, whose presence is inherent to the assembly process"*.

More definitions for assembly features are presented in [19].

In this paper, assembly features are used to integrate product design and APP. In order to interface product design knowledge in solid/assembly modelers with downstream application within the product-life cycle, two complementary approaches have been used: design by feature and feature recognition. In the following subsections, a review for the two approaches is presented.

A. Design by Feature (Feature-Based Model)

Feature-based modelling was developed [10], [13] in order to fill the gap between detailed geometry information, the elementary relations, and abstract functional information [16]. Feature-based modelling for assembly has been developed [12] to support the flow of information from a CAD model into downstream applications such as APP.

Many researchers use assembly modelling based on features either for representing assembly knowledge or for assembly planning. Deneux [18] describes how assembly features have been used in the planning process for aircraft assembly. Bronsvoort and Van Holland [17] use assembly features in product modelling and planning (grip planning, motion planning and assembly sequence planning). Van Holland [3] describes assembly features as *''features with significance for assembly processes''* and which are composed of connection features and handling features. Connection features capture the connection between components and handling features capture the handling of components by grippers during assembly. They propose the concept of ''Related'' and ''Relation'' as the base classes for both part and assembly modelling. Shyamsundar and Gadh [21] introduce a representation of the assembly (called the AREP), which stores assembly hierarchy as well as relations between components and sub-assemblies. In AREP, relations are defined in terms of assembly features which are classified into relational assembly features and assembly form features. The former indicates the relation between geometric features, and the latter indicate the joining of two form features from different parts.

Singh and Bettig [22] propose assembly ports to group and automate mating relations between parts. An assembly port is distinct from assembly feature in the sense that an assembly port is the low-level geometrical entity (ex. peg), which will be connected to another assembly port (ex. hole) through an assembly feature (peg-hole connection). Their work presents an attempt to separate geometrical information (assembly port) from assembly information (assembly feature) but there is no clear structure for the low level geometrical entities that form the assembly port. Semantic feature modelling is presented by Bidarra and Bronsvoort [24]. In their model, a feature is described over well-defined meanings (semantics), which are specified in classes. The classes form a structure of instances of all properties of a given feature type. Another semantic assembly modelling approach is presented by Hui et al. [25], where assembly information is described by a three-level semantic abstraction: conceptual level, structural level, and part level.

In the following and as a conclusion from this brief survey, the below points have been taken into consideration in modelling assembly features for process planning:

1) Feature-based modelling for assembly has been used to improve the efficiency of assembly modelling and assembly planning. It provides a considerable help in integrating assembly design and assembly planning.
2) Assembly feature semantics should include geometrical, non-geometrical, and functional properties (during assembly) as well as assembly relations.
3) It is preferable to separate geometrical knowledge representation from assembly knowledge to give more options for designers in terms of changing geometry while keeping other assembly information.

B. Feature Recognition

The term Feature recognition (FR) refers to the different techniques that are used to extract the knowledge enclosed in geometrical and assembly representations of solid models in order to be used in manufacturing, process planning, and other different downstream applications in the product life-cycle. During the last decades, a lot of work has been published towards effective and smart automatic feature recognition [20] and several different methods for feature recognition have been reported [23].

From a geometric point of view, feature representation is classified into two types: surface-based or volume-based. Surface features are based on topological entities such as face, edges, and vertices with functional meanings on the part boundary; this representation is known as boundary representation (B-rep). Volumetric features are based on three-dimensional geometrical primitives such as sphere or cylinder, and this representation is known as constructive solid geometry (CSG). Based on this classification, Feature recognition approaches can be classified as well into B-rep-based approaches and CSG approaches. Since the B-rep CAD representation of features is widely and mostly used, the B-rep feature recognition approaches are the most popular in published literature [26].

From technical and engineering points of view FR systems are divided according to two methods; external and internal methods. In internal methods, the Application Programmable Interface (API) of the CAD software is used in order to extract topological, geometrical and assembly information relating to the part or assembly. While in external methods, the CAD file model is exported in a neutral data format (STEP, IGES, ACIS or similar), and then translated to be suitable for feature extraction [27]. Both methods have been used by researchers for feature recognition purposes.

In this paper, an internal B-rep CAD recognition approach is proposed and this approach will be further illustrated in the next section.

III. Proposed Approach

The proposed approach is briefly sketched in Fig. 1. The three stages of recognition, modelling, and integration are illustrated. In the first stage, geometrical knowledge represented by form features and assembly knowledge represented by mating features are recognized from the CAD models. An intermediate modelling stage follows where the recognized geometry and assembly knowledge are modelled in a form feature semantic model and a mating surface model, respectively. The last stage will be to structure and store the recognized assembly and geometric semantic knowledge according to a well-defined ontology in order to facilitate sharing and integrating this knowledge with the APP.

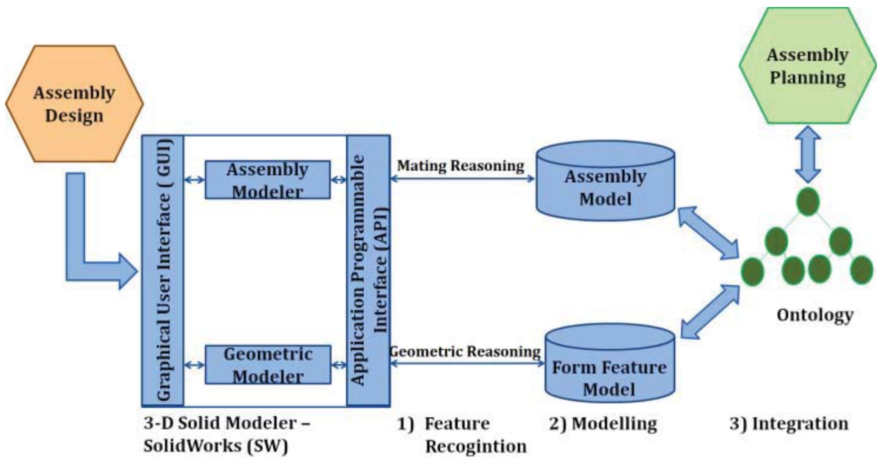

Fig. 1 Proposed approach for integrating assembly design and assembly planning

The FR strategy is based on the following steps:

1) Extracting the related geometry knowledge enclosed in the geometry representation of the part/assembly model as a B-rep model. B-rep modelling decomposes a solid into its boundary surfaces or shell. Each shell can be decomposed in to individual faces. Each face is described as a surface bounded by a loop of edges. Each edge bounded by two vertices. The data structure of the B-rep model is described by topological entities and geometrical entities; topological entities include shells, edges, co-edges, faces, loops, and vertexes. Geometrical entities include curves, points, and surfaces. The topology serves to describe the structure of the model, while the geometry describes the shape [28].
2) Extracting non-geometrical semantic knowledge related to form features in each part in the assembly from the geometric/part modeler. The non-geometrical semantic knowledge includes dimensional and positional knowledge. Dimensional knowledge includes geometrical tolerance and dimensions (GT&D). Dimensions (width, height, diameter etc.) and geometrical tolerances (line profile, surface profile, surface finish etc.) assigned for each geometrical and topological entity will be extracted. Positional knowledge is needed in order to determine the orientation of the form feature in the three-dimensional space. According to [32], positional information includes a reference point existing within the geometrical boundaries of a form feature and a reference line passing through the reference point. The reference point requires its positional x, y, and z coordinates, while the reference line is defined through its angles relative to x, y, and z axes.
3) Extracting assembly knowledge relations enclosed in mating features from the assembly modeler. Parts are added to an assembly by specifying mating features or constraints. Mating features are used between assembly parts to constrain their degrees of freedom to correct locations/orientations; they can offer information about parts locations/orientations and their connectivity relationship in the assembly. The B-rep model along with the extracted mating feature will be used to create the mating surface model.

The three-step extraction procedure is illustrated by a three-part assembly example (Fig. 2). The form feature semantic model is illustrated in Fig. 3, while the mating surface model is presented in Fig. 4.

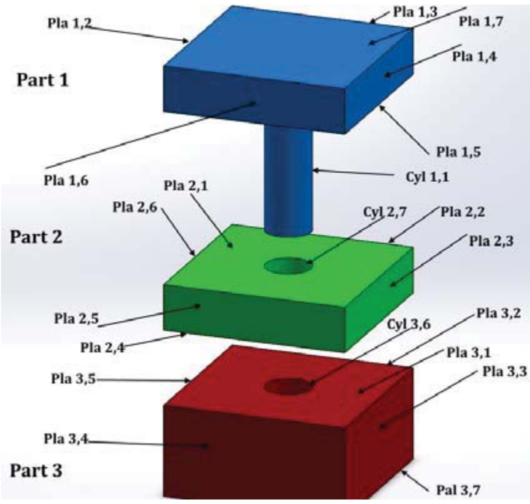

Fig. 2 Three-part assembly

In Fig. 2, a three-part assembly (rectangular head peg and two cubes with through-holes) is presented. The surfaces for each part are indicated, where Pla stands for planar surface, Cyl for cylindrical surface.

The form feature semantic model for the 3-part assembly is illustrated in Fig. 3. A five- layer semantic model is presented, where the first layer is for the compound features (peg with head, cube with through hole), which are composed of primitive features (boss, hole). The primitive features are composed of the B-rep entities (geometrical and topological) like surfaces, profile, centerlines, and so on. The last two layers are for position/orientation and geometrical dimension and tolerances (GD&T). The position layer consists of reference line and reference point for each compound feature, while the GD&T layer consists of dimensions and tolerances for the B-rep entities.

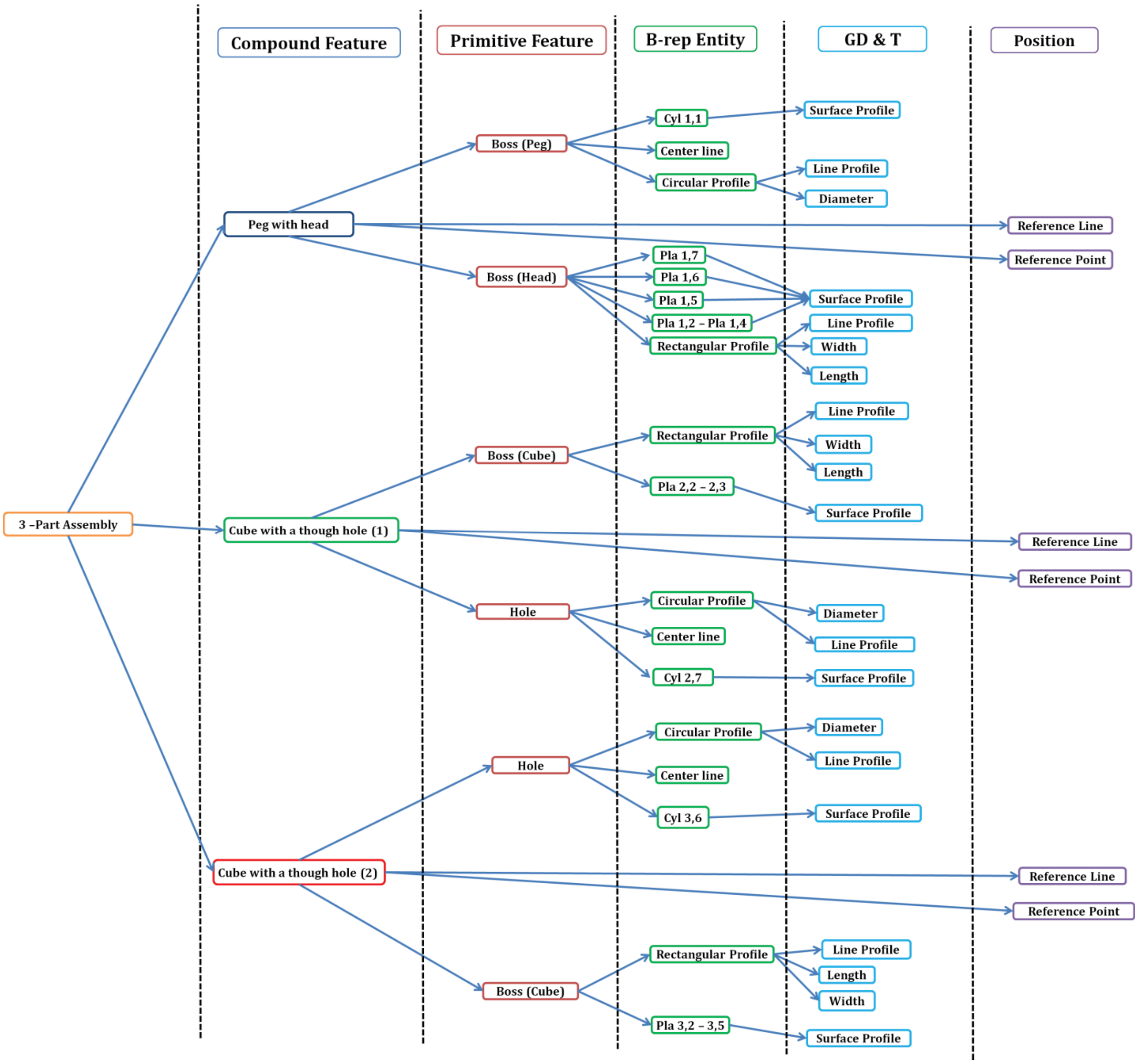

Fig. 3 Form feature semantic model for 3-part assembly

The mating surface model for the 3-part assembly is presented in Fig. 4 where the mating surfaces are connected via mating relations. Two types of mating relations are presented: A coincident relation type between planar surfaces (Pla 1,5 – Pla 2,1 and Pla 3,1- Pla 2,4), and a concentric relation type between cylindrical surfaces (Cyl 1,1 – Cyl 2,7 and Cyl 1,1 – Cyl 3,6). In published literature, the term "mating feature" has been used to describe the mating relations between planar surfaces [3], while "alignment feature" has been used to represent the mating relation between cylindrical surfaces, where the alignment of the axes of the two cylindrical surfaces in an assembly is defined [30].

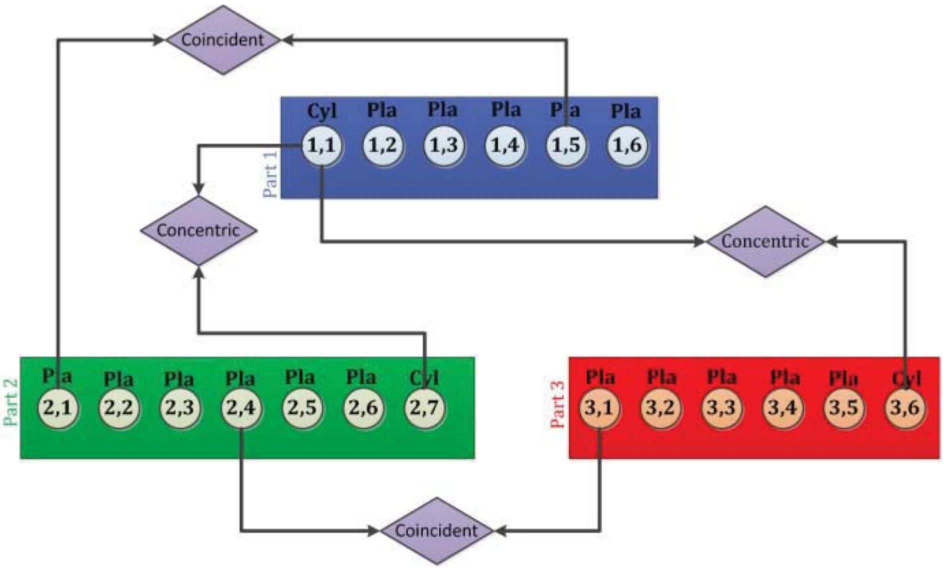

Fig. 4 Mating surfaces model for 3-part assembly

Both models in Figs. 3 and 4 will be used for determining the required assembly processes and resources. According to Smale and Ratchev [29], the basic core of assembly processes is "Moving Part x" and "Joining Parts x and y". In order to determine assembly processes, more specialized forms of assembly features are required. *Handling features*, which in this case is a generic form of assembly information (independent of the actual position and orientation of the component within an assembly), will be used to store and retrieve information about moving (transporting) processes such feeding, fixturing, and grasping [3]. While handling processes (grasping, feeding, fixturing) take place on the surfaces, which are not included in any mating relations (non-mating surfaces), joining processes will take place on mating entities (surfaces, edges). Therefore, further specialization is required for mating features in order to determine the "joining features". The concept of joining feature has been proposed [31] to represent assembly/joining relations, and it includes joining entities, joining methods, constraints and groove shapes.

In this paper, the *joining feature* concept will be expanded to include geometrical, dimensional and positional characteristics of the mating entities, which give a valuable aid in determining joint types and the required joining processes to join those mating entities. Fig. 5 shows an assembly structure model based on the previous semantic form model (Fig. 3) and mating surface model (Fig. 4), where a five- layer structure is proposed to give a full description of generating joining features based on geometrical, dimensional and positional information of form features.

In Fig. 5, the first and the second layers are for assembly-part structure knowledge in the product design. Each product is composed of several subassemblies, and each subassembly is composed of at least two parts. The third layer is a feature layer, where two associated form features from different parts are connected via an assembly feature. The feature layer represents the feature-based assembly modelling for the product. The next layer represents the B-rep model extracted from the geometric/part modeler. In the next layer the assembly knowledge and the form feature semantic knowledge are both deployed to generate more specialized application-specific features. Surfaces of the B-rep model are classified into mating surfaces represented by mating (for planar surfaces) and alignment (for cylindrical surfaces) features, and non-mating surfaces, some of them are represented by handling features. Handling features will be further specialized into gripping, feeding, and fixturing features for each part in the assembly.

In order to determine joining features, mating surfaces are further analyzed by determining attributes of the contact area between mating entities, in order to capture information related to potential joining processes.

An example for contact area attributes are groove face, root edge, root opening and root angle (Fig. 6). The identification of those attributes facilitates the determination of the required joining process. For the attributes illustrated in Fig. 6, the joining process will be welding, and the contact area attributes will be known as welding features.

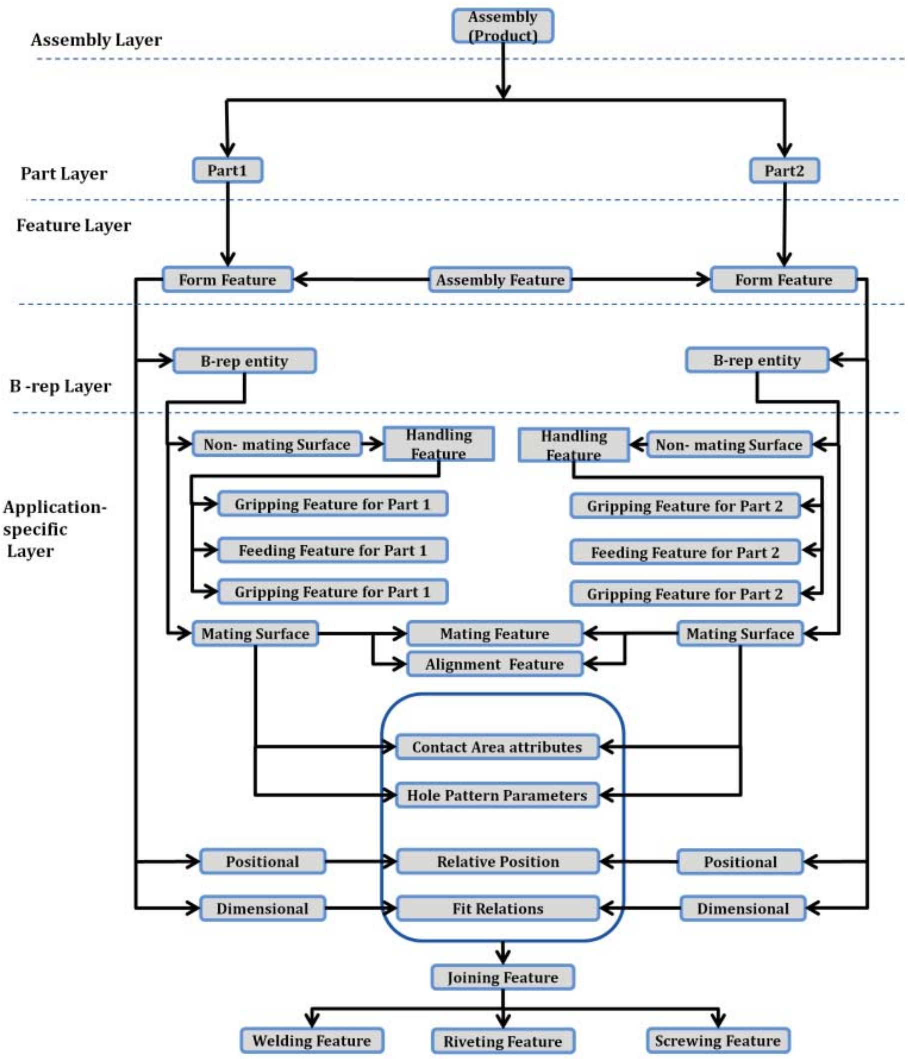

Fig. 5 Assembly semantic model

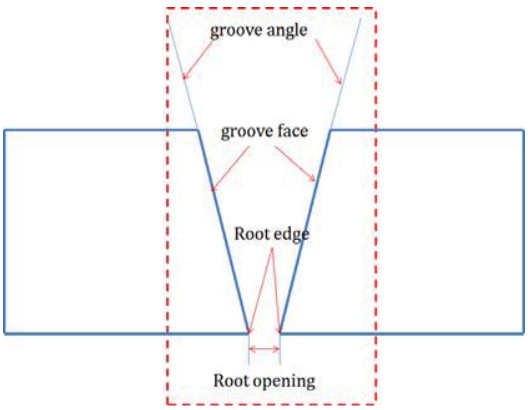

Fig. 6 Contact area parameters for welding process

Another important information that could be identified from mating surfaces is the hole pattern. A hole pattern has several attributes that could help to determine a specific joining process. One of these attributes is the hole type. Identification of a hole as threated determines screwing as a joining process to follow.

Positional and dimensional information of form features are important for determining the required joining processes. Positional information helps to determine the relative positions of the form features, where for example overlapping between the mating entities, with unthreaded holes, indicates riveting as a joining process.

Dimensional parameters of form features will aid in determining fit relations between mating surfaces. Three types of fit relations are mentioned in literature, clearance fit between hole and shaft, which is identified if the minimum allowable dimension of a hole is larger than the maximum allowable dimension of a shaft. Transition fit, which is identified if the minimum allowable dimension of a hole is smaller than the maximum allowable dimension of a shaft, and the maximum allowable dimension of a hole is larger than the minimum allowable dimension of a shaft. The last type of fit is interference fit, which is identified if the maximum allowable dimension of a hole is smaller than the minimum allowable dimension of a shaft. Identifying fit relations will aid in determining the type of fit process whether its press

International Science Index, Mechanical and Mechatronics Engineering Vol:10, No:10, 2016 waset.org/Publication/10005518

fitting (for interference fit) or shrink fitting (for other fit types).

The extracted semantic data will be shared and integrated with APP via an ontology, which represents the third and the final stage of the proposed approach. Ontology development will be discussed in the next section.

IV. Ontology Development

An ontology can be regarded as "a data model that represents a domain and is used to reason about the objects in that domain and relations between them" [33]. Ontologies specify "a domain-specific vocabulary of entities, classes, properties, predicates, and functions, and a set of relationships that necessarily hold among those vocabulary items" [34]. Ontologies have been used to capture and share product design knowledge [35], to integrate engineering applications and to solve interoperability problems.

In this paper, a three-layered architecture of engineering ontologies in product design and APP is proposed (Fig. 7). The proposed layered ontology structure consists of:

1) General Foundation Ontology (GFO)
2) Domain Specific Ontology (DSO)
3) Application Specific Ontology (ASO)

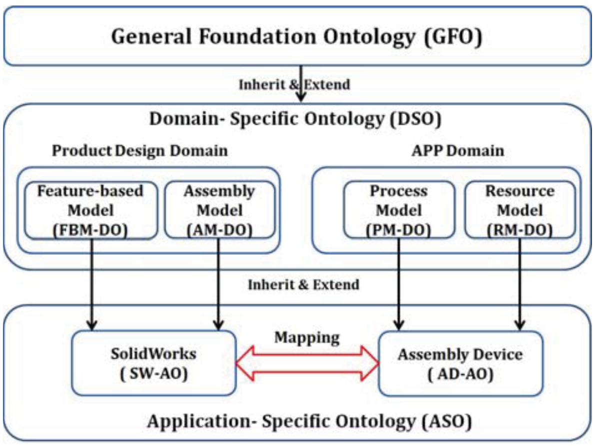

Fig. 7 Three-layered architecture of engineering ontologies in product design and APP

The GFO is the first upper layer ontology, which is designed to provide common concepts, such as *product*, *feature*, *material*, *process*, and *resource* which are inherited by the DSOs such as FBM-DO, AM-DO, PM-DO and RM-DO. The Domain Specific Ontologies represent the second level of the proposed architecture; those ontologies will add domain-specific concepts which belong to that particular domain. The third level is the ASOs (such as SW-AO and AD-AO); those ontologies will capture semantics specific to each application. Two applications have been included: SolidWorks as product design application and assembly robotic device (ex. high speed assembly robot Sony SRX series) as APP application. The knowledge transfer between different ASOs can be accomplished through mapping procedures which discovers similar or matching concepts and properties.

All of the ontologies are implemented by using the Protégé-OWL editor. In the following subsections; the three different ontologies of the layered ontology structure will be discussed.

A. General Foundation Ontology (GFO)

Foundation ontologies consist of generic, abstract, and high level concepts which can be applied to a wide range of domains. Foundation ontologies also provide a knowledge base for more specialized ontologies [36].

The GFO contains the general key concepts, which are common and applied to any of the domains in product design and APP. The concepts defined in GFO are product, feature, material, process, and resource. Fig. 8 shows concept definitions and attributes for the GFO.

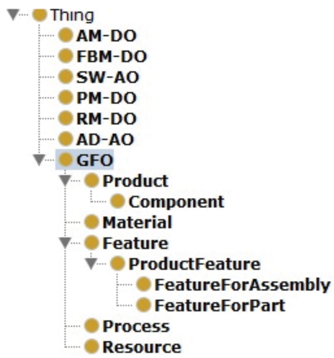

Fig. 8 Class hierarchy and the definition of concepts of the GFO

The concepts in the GFO ontology have attributes, which will be inherited by the different domain ontologies. For example, the *Component* subclass, which describes the basic structural design entity under the product class, will be further inherited by FBM-DO and SW-AO. FBM-DO will further embody Component with *Assembly*, *Subassembly* and *Part* subclasses. The same will be applied to the Feature class and its subclasses: *FeatureForPart* and *FeatureForAssembly*, which will be further inherited by the FBM-DO and SW-AO. The properties defined in FGO are: *is-a*, *is-a-part-of*, *is-composed-of* and *has-attribute-of*. The first two properties reflect the inheritance relations between different concepts. The last two properties define the relations between concepts and its attributes.

B. Domain Specific Ontology (DSO)

The DSO layer consists of four domain ontologies (DO). Two of those are in the product design domain, namely the Feature-based Model (FBM-DO), and the Assembly Model (AM-DO), and the other two are in the APP domain, namely Process Model (AM-DO) and Resource Model (RM-DO). Each DO reuses concepts and properties from the FGO and defines more specified, expanded and specialized concepts/ properties for a particular domain.

The FBM-DO is created to capture knowledge about a product's structure and form domain. Fig. 9 illustrates the class hierarchy and the definition of concepts of the FBM-DO.

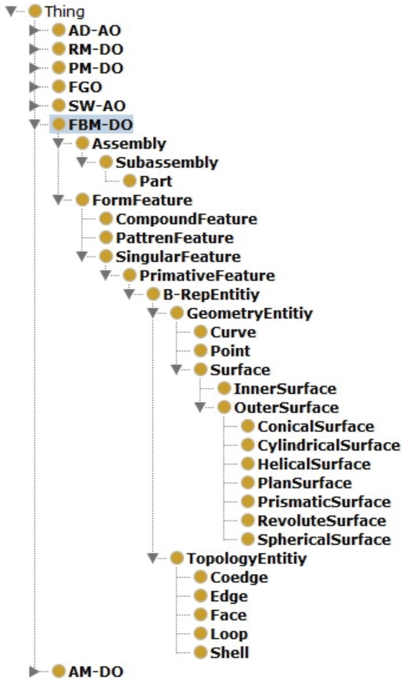

Fig. 9 Class hierarchy and the definition of concepts of the FBM-DO

In Fig. 9, the FBM-DO expands the product structure and geometry based on feature modelling. *Assembly*, *Subassembly* and *Part* classes represent the product basic structure, where the Subassembly is composed at least of two parts. The *Part* class is further decomposed into its features. Each part is composed at least of one form feature. The *FormFeature* class is decomposed according to complexity into: *PatternFeature*, *SingularFeature*, and *PrimitiveFeature*. *PrimitiveFeature*, which is considered as the basic form feature unit is further decomposed into *B-RepEntity* class, which will be decomposed further into the very basic geometrical and topological entities: *GeometryEntity* and *TopologyEntity*. *GeometryEntity* has attributes *Surface, Curve and Point.* The *Surface* class includes all different types of surfaces used in geometric modelers. *TopologyEntity* has attributes *Edge*, *Shell*, *Loop*, *Face*, and *Co-edge*.

AM-DO is created for assembly modelling as part of the product design domain. Fig. 10 illustrates the class hierarchy and the definition of concepts of the AM-DO.

If the FBM-DO represents the form attribute (geometrical and structural information) of the product design, AM-DO represents the behaviour of the design unit during assembly. AM-DO includes three major subclasses: *SpatialRelationship*, *DegreeOfFreedom*, and *AssemblyFeature*. *SpatialRelationship* expresses the relative positions of parts in an assembly in their final state. *DegreeOfFreedom* is used to describe the motion (translation and rotation) of parts during assembly. The third subclass, *AssemblyFeature*, is composed of *Mating*, *Alignment*, *Handling*, *Joining*, and *Tooling* features. The *AssemblyFeature* class introduces necessary assembly design information to establish a link with assembly processes and resources for APP. While joining features, with its further specialization (welding features, fastening features etc.), represent a link for integration with joining processes. Handling and tooling features represent a link for integration with assembly resources. Handling features represent the geometrical characteristics of the part that are needed to determine the required assembly transporting resources such as fixture, feeder, and gripper. Tooling features represent the geometrical characteristics of the part's shape that are needed to determine the required assembly tooling resources. An example of the tooling features is the shape and the size of the screw's head, which are required to determine the suitable wrench to fasten this screw.

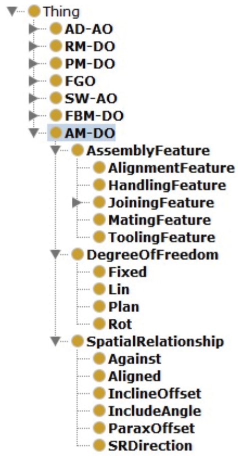

Fig. 10 The class hierarchy and the definition of concepts of the AM-DO

The next two DSOs are the PM and the RM of the APP domain. The PM-DO is illustrated in Fig. 11, where the *process* class in GFO is expanded and inherited by PM-DO into *AssemblyProcess* and *ManufacturingProcess* classes. The *AssemblyProcess* class is further expanded into *JoiningProcess* and *HandlingProcess* classes. The *JoiningProcess* class is composed of subclasses representing different joining processes in APP such as *Welding* and *Fastening*. The *HandlingProcess* class is composed of *Gripping*, *Feeding*, and *Fixturing* subclasses.

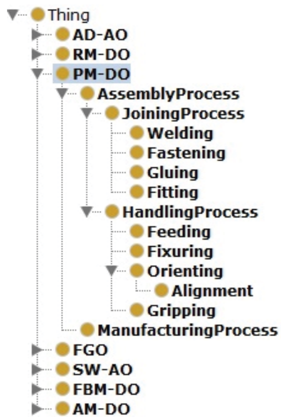

Fig. 11 The class hierarchy and the definition of concepts of the PM-DO

The RM-DO represents manufacturing and assembly resources in APP (Fig. 12). The *AssemblyResource* class is further decomposed into several subclasses according to complexity from *Enterprise* and *Factory* subclasses into *Area*, *Line*, *Cell*, *DeviceCombination*, and *IndividualDevice*.

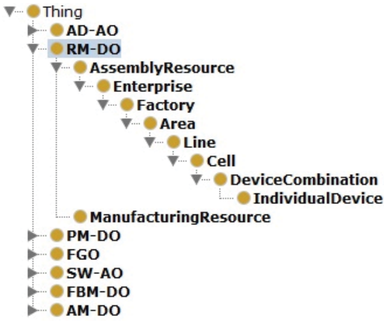

Fig. 12 The class hierarchy and the definition of concepts of the RM-DO

The *IndividualDevice* subclass is further inherited by AD-AO in the ASO layer, which will be discussed in the next section.

C. Application Specific Ontology (ASO)

So far, the ASO represents the lowest/ level of the proposed ontology. ASO defines more specified, expanded and specialized concepts/properties for a particular application. ASO is used to transfer product data semantics between different engineering applications. In this paper, two ASOs are developed: SW-AO to share product design data semantics from SolidWorks CAD software, and AD-AO to utilize assembly processes and resources in converting product data semantics into an assembly process plan for performing assembly of a finished or semi-finished product.

SolidWorks, as a commercial product design package, has been widely used as a 3-D geometrical modeler in various product life-cycle and product development applications. SW-OA (Fig. 13) inherits and expands concepts from ontologies at higher levels such as *FormFeature* from FBM-DO and *MatingFeature* from AM-DO. For example, *FormFeature* from FBM-DO inherits and expands into *Round, Revolve, Hole, Fillet, Extrude*, and *Chamfer* under *ShapeFeature* class in the SW-AO. *MatingFeature* from AM-DO inherits into *Concentric, Tangent, Perpendicular, Parallel and Coincident* under *AssemblyConstriants* in the SW-AO. SW-AO also defines unique concepts, which are only used in SW. An example of the unique classes in SW-AO is the *DimXpertManger*. This class is composed of several subclasses such as *ReferenceManger*, *GeometricTolerance*, and *Dimensions*. The *ReferenceManger* subclass determines positional parameters of the features. Data for lines and points have been determined under *Datumline* and *DatumPoint*, respectively. The two subclasses *GeometricTolerance* and *Dimensions* include all different types of dimensions and tolerances, which have a direct impact on geometrical variations in the assembly design.

The SD-AO represents robotic assembly devices and consists of several units, which are represented by subclasses: *HandlingAndOrientingTools*, *JoiningTools*, *ToolChanger* and *Robot.* The first two subclasses include all different tools that will be used in handling, orienting and joining parts during assembly. The *FixturingTool*, *GrippingTool* and *FeedingTool* are subclasses for the *HandlingAndOrienting* class. Different types of gripping tools as *PincerGripper, MagnetGripper, VacuumGripper* and *FingerGripper* under *GrippingTool* subclass. Attributes and properties could be defined for each gripper type such as gripping range, gripping power and force. *JoiningTools* includes *WeldingTool*, *PressingTool,* and *ScrewingTool.*

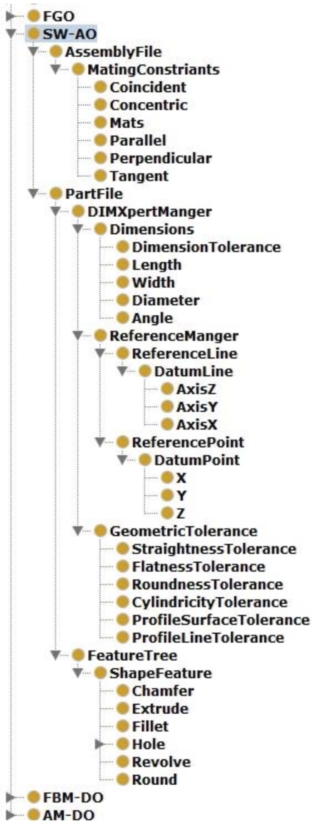

Fig. 13 The class hierarchy and the definition of concepts of the SW-AO

The Robot class includes different robots that are commonly used in robotic assembly devices such as ScaraRobot, MobileRobot, and HexapodRobot.

The integration between product design domain and APP will be performed through a mapping procedure between SW-AO and SD-AO. The processes and resources represented by different tools in SD-AO will be selected according to the product design semantics represented in SW-AO. For example, a width dimension in Dimensions class in SW-AO may determine the type of gripping (whether it is finger gripping or magnet gripping) in SD-AO. Another example is that a type of a hole in ShapeFeature class in SW-AO might determine the joining tool in SD-AO.

The ontology part in this paper will be expanded in further work by defining axioms for the FDO and properties for the DSOs and ASOs. Also a detailed mapping procedure based on defined properties of SW-AO and AD-AO has to be performed in the future work.

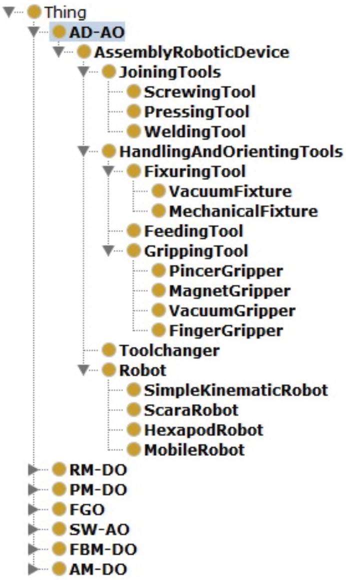

Fig. 14 The class hierarchy and the definition of concepts of the SD-AO

V. Conclusion

In this paper, an approach for extracting product design semantics for assembly and integrating those semantics with APP is proposed. The approach is based on internal B-rep recognition for extracting product design semantics, and on a three-layered architecture of engineering ontologies for sharing and integrating product design semantics with APP. Future work includes upgrading and expanding the ontology part by developing the DSO and ASO layer. This development includes expanding and defining properties for classes in various ontologies. For the ASO layer, mapping procedures have to be developed for further integration between product design domain and APP domain.

References

[1] S. Cho, Mechanical assembly. The mechanical engineering handbook Series, CRC Press LLC; 2005.
[2] Du H. Zha XF, "A PDES/STEP-based model and system for concurrent integrated design and assembly planning" Computer Aided Design, 2002, 34: pp. 1087-1110.
[3] W. Van Holland, Assembly features in modelling and planning, PhD thesis, Delft: Delft University of Technology; 1997.
[4] W. Bronsvoort, W. Van Holland and F. Jansen, Feature modelling for assembly. Graphics and Robotics, Springer-Verlag, 1995, pp. 131-148.
[5] L. Wang, S. Keshavarzmanesh, H-Y. Feng and R. Buchal, "Assembly process planning and its future in collaborative manufacturing: a review", The International Journal of Advanced Manufacturing Technology, 2008, 41(1-2): pp. 132–144.
[6] L. Michael, Planning and scheduling in manufacturing and Services, Springer, 2009.
[7] C. Dartigues, P. Ghodous, M. Gruninger, D. Pallez and R. Sriram, "CAD/CAPP Integration using Feature Ontology. Concurrent Engineering", 2007, 15(2): pp. 237–249.
[8] LS. Wierda, "Linking Design, Process Planning and Cost Information by Feature Based Modelling", Journal of Engineering Design, 1991, 2(1): pp. 3-19.
[9] L. Wingard, Introducing Form Features in Product Models, A step Towards CAD CAM with Engineering Terminology, Licenciate Thesis, KTH: Royal Institute of Technology, 1991.
[10] J. Shah and M. Rogers, "Functional requirements and conceptual design of the feature-based modelling system", Computer-Aided Engineering Journal, pp. 9-15, 1988.
[11] OW. Salomons, FJ. Houten, A. Van, HJ. Kals, "Review of Research in Feature-Based Design", 1993; 12(2): pp. 113–132.
[12] J. Shah and M. Rogers, "Assembly modeling as an extension of feature-based design", Research in Engineering Design; 5: pp. 218–37, 1993.
[13] J. Shah and M. Rogers, 1989, "Geometric modeling for automated manufacturing", Proceedings of the 16th NSF Design & Manufacturing Systems Grantees Conference, 1989, pp. 371-375.
[14] CK. Chan and ST. Tan, "Generating assembly features onto split solid models", Computer-Aided Design, 2003, 35(14): pp. 1315-1336.
[15] O. Coma, C. Mascle and P. Véron, "Geometric and form feature recognition tools applied to a design for assembly methodology", Computer-Aided Design, 2003, 35(13): pp. 1193-1210.
[16] M. Lanz, Approach to feature-based modeling and analysis, Master thesis, Tampere University, 2004
[17] W. Bronsvoort and W. Van Holland, "Assembly features in modeling and planning", Robotics and Computer Integrated Manufacturing 2000, 16: pp. 277–94.
[18] D. Deneux, ''Introduction to Assembly Features: An Illustrated Synthesis Methodology", Journal of Intelligent Manufacturing, 1999, 10: pp. 29–39.
[19] H. Ullah, E. Bohez and M. Irfan, "Assembly Features: definition, classification, and usefulness in sequence planning", International Journal of Industrial and Systems Engineering, 2009, 4(2): pp.111-132.
[20] B. Babic, N. Nesic and Z. Miljkovic, "A review of automated feature recognition with rule-based pattern recognition", Computers in Industry, 59 (4): pp 321-337.
[21] N. Shyamsundar and R. Gadh, "Internet-based collaborative product design with assembly features and virtual design spaces", Computer-Aided Design, 2001; 33: pp. 637–51.
[22] P. Singh and B. Bettig, "Port-compatibility and connectability based assembly design", Journal of Computing and Information Science in Engineering" 2004, 4:197 -205.
[23] J. Shah, D. Anderson, Y. Kim and S. Joshi, "A Discourse on Geometric Feature Recognition from CAD Models", Journal of Computing and Information Science in Engineering, 2001, pp. 41-51.
[24] R. Bidarra and WF. Bronsvoort, "Semantic feature modelling", Computer-Aided Design, 2000, 32(3): pp. 201-225.
[25] W. Hui, X. Dong, D. Guanghong and Z. Linxuan, "Assembly planning based on semantic modeling approach. Computers in Industry", 2007, 58(3): pp. 227-239.
[26] V.B. Sunil, R. Agarwal and S.S. Pande, "An approach to recognize interacting features from B-Rep CAD models of prismatic machined parts using a hybrid (graph and rule based) technique", Computers in Industry, 2010, 61(7) pp. 686-701.
[27] H. K. Miao, N. Sridharan, and J. J. Shah, "CAD-CAM integration using machining features", International Journal of Computer Integrated Manufacturing, 2002, 296–318.
[28] I. Stroud, Boundary Representation Modelling Techniques, 2006, London: Springer.
[29] D. Smale, A. Ratchev, "Capability Model and Taxonomy for Multiple Assembly System Reconfigurations", 13th IFAC Symposium on Information Control Problems in Manufacturing, Moscow, 2009, pp. 1923-1928.
[30] J. Shah, "Designing with Parametric CAD: Classification and Comparison of Construction Techniques", F. Kimura (Ed.), Geometric Modelling: Theoretical and Computational Basis Towards Advanced CAD Applications, 2001, pp.53-68.
[31] K. Kim, Assembly operation tools for. Assembly operation tools for e product design and realization, PhD thesis, Pittsburgh: University of Pittsburgh, 2003.
[32] N. A. Anjum, "Verification of Knowledge Shared Across Design and Manufacture Using Foundation Ontology". PhD Thesis: Loughborough University, 2011.
[33] T. R. Gruber, "Toward Principles of the Design of Ontologies Used for Knowledge Sharing", International Workshop on Formal Ontology in Conceptual Analysis and Knowledge Representation, 1993.

[34] R. Fikes and A. Farquhar, "Distributed Repositories of Highly Expressive Reusable Ontologies", IEEE Intelligent System, 1999, 14(2): pp. 73-79.

[35] L. Patil, D. Dutta, and R. Sriram, "Ontology-Based Exchange of Product Data Semantics", IEEE Transactions On Automation Science and Engineering, 2005, 2(3): pp. 213-225. S. Chen, B. Mulgrew, and P. M. Grant, "A clustering technique for digital communications channel equalization using radial basis function networks," *World Academy of Science, Engineering and Technology Trans. Neural Networks*, vol. 4, pp. 570–578, July 1993.

[36] S. Sanchez-Alonso and E. Garcia-Barriocanal, "Making use of upper ontologies to foster interoperability between SKOS concept schemes", Online Information Review, 2006, 30(3): pp. 253-277.