A Model for Capturing Product Assembly Information

Sudarsan Rachuri,Young-Hyun Han, Sebti Foufou[1], Shaw C. Feng,
Utpal Roy[2], Fujun Wang, Ram D. Sriram, Kevin W. Lyons

Design Process Group, Manufacturing Systems Integration Division, NIST, Gaithersburg, MD 20899, USA. email: {sudarsan, sfoufou, sfeng, sriram, klyons}@cme.nist.gov

[1] Lab. Le2i, Université de Bourgogne, BP 47870, 21078 Dijon, France. Currently guest researcher at Manufacturing Systems Integration Division, NIST, Gaithersburg, MD 20899, USA. email: sfoufou@u-bourgogne.fr

[2] Department of Mechanical, Aerospace and Manufacturing Engineering, Syracuse University, Syracuse, NY 13244, USA. email: uroy@ecs.syr.edu

Abstract:
The important issue of mechanical assemblies has been a subject of intense research over the past several years. Most electromechanical products are assemblies of several components, for various technical as well as economic reasons. This paper provides an object-oriented definition of an assembly model called the Open Assembly Model (OAM) and defines an extension to the NIST Core Product Model (NIST-CPM). The assembly model represents the function, form, and behavior of the assembly and defines both a system level conceptual model and associated hierarchical relationships. The model provides a way for tolerance representation and propagation, kinematics representation, and engineering analysis at the system level. The assembly model is open so as to enable plug-and-play with various applications, such as analysis (FEM, tolerance, assembly), process planning, and virtual assembly (using VR techniques). With the advent of the Internet more and more products are designed and manufactured globally in a distributed and collaborative environment. The class structure defined in OAM can be used by designers to collaborate in such an environment.

The proposed model includes both assembly as a concept and assembly as a data structure. For the latter it uses STEP. The OAM together with CPM can be used to capture the assembly evolution from the conceptual to the detailed design stages. It is expected that the proposed OAM will enhance the assembly information content in the STEP standard. A case study example is discussed to explain the Usecase analysis of the assembly model.

Keywords: Assembly modeling, UML, kinematics representation, Assembly features, Standards, STEP

List of Figures

Figure 1: Class Diagram of the Open Assembly Model.
Figure 2: Tolerance Model.
Figure 3: Exploded View of the Planetary Gear Model.
Figure 4: Artifact Associations in the Planetary Gear System.
Figure 5: Ring Gear Assembly.
Figure 6: Instance Diagram of Ring Gear Assembly.
Figure 7: Planet Carrier Assembly.
Figure 8: Instance Diagram of Planet Carrier Assembly.
Figure 9: Planet Gear Carrier Assembly.
Figure 10: Instance Diagram of Planet Gear Carrier Assembly.
Figure 11: Planet Gear Carrier and Sungear Assembly.
Figure 12: Instance Diagram of Planet Gear Carrier and Sungear Assembly.
Figure 13: Planet Gear Carrier and Ring Gear Assembly.
Figure 14: Instance Diagram of Planet Gear Carrier and Ring Gear Assembly.

List of tables

1. Introduction

The design of complex engineering systems is increasingly becoming a collaborative task among designers or design teams that are physically, geographically, and temporally distributed. The complexity of modern products is such that a single designer or design team can no longer manage the complete product development effort. Designers are no longer merely exchanging geometry data, but also more general knowledge about design and the product development process, including specifications, design rules, constraints, rationale, etc. Furthermore, this exchange of knowledge more and more often crosses corporate boundaries. As design become increasingly knowledge-intensive and collaborative, the need for computational frameworks to support product engineering in industry becomes more critical. Though Computer Aided Design (CAD) vendors have developed many different ways to model parts and represent design information as constraints between parts, it is not clear that all of these representations are capturing the same level of information. The issue of exchanging parts and assembly information between modeling systems is critical for unrestricted exchange of product data. However, little has been done in terms of developing standard representations that specify assembly information and knowledge. An assembly information model contains information regarding parts and their assembly relationships. Hence, we wish to emphasize the nature and information requirements for these part features and for these assembly relationships. Further we need to address the evolution of their corresponding information models during the conceptual and detailed design stages. In this paper, we propose an integrated information model for assembly representations.

This is important for the exchange of information between modeling, analysis and planning systems. The paper is organized as follows. We start with a brief review of the current work in assembly representation of products (in Section 2). We present the object-oriented representation of electro-mechanical assemblies using Unified Modeling Language (UML) [1] in Section 3. In Section 4, we discuss a Usecase analysis of the assembly model. Finally, the conclusions and further research work are presented in Section 5.

2. Previous Work and Current Status

ISO 10303-Part 44 [2] provides for some limited assembly design representations that capture the assembly structure and the kinematic joint information. The assembly model presented here establishes a neutral representation of assemblies of products, which are composed of sets of components. In this model, the complete products are called assemblies, and the components at the lowest levels in the assemblies are called parts. The model focuses on the hierarchy of the product, and on the position and orientation between parts. One of the primary features defined in ISO 10303 is that it provides a data modeler to generate various types of product data structures (e.g., Bill-Of-Materials (BOM), parts list, etc.,) using the same primitive entities. However, it should be noted that ISO 10303 does not adequately address the following: (1) The relationship among different product definitions for the same product (e.g., the relationship of a product definition for a component in a preliminary design to a corresponding product definition for the same component in a detailed design is not captured), (2) The change process for a product including the reasons for the change, and (3) The decisions made and their rationale, for the entire product life cycle. ISO working group (TC 184/SC4/WG12) [3] has proposed to enhance the STEP's assembly representation. In their proposal, they have defined detailed geometric information not only for hierarchical relationship but for peer to peer relationships among component parts via assembly features. Geometric constraints among component parts at the detailed geometric element level are also enabled. They have included more information on component association and included detailed information about appropriate assembly features involved in component associations. It should be noted that the ISO proposal does not cover configuration management of assemblies and components. Although the proposal outlined the possible applications of the proposed assembly representation in four areas: kinematic analysis of assemblies; animation of assemblies; assembly/disassembly process planning; and tolerance analysis and synthesis, the actual application methodologies were not identified or reported. For a detailed description of a feature based CAD/CAM system, [4] is an excellent reference.

NIST has been actively involved in identifying and developing representational methodologies for the next generation of assembly-related standards. Our research seeks ways to assist designers with assembly considerations throughout the different phases of the complete product realization process, from conception to assembly analysis and final process plan development. Readers are encouraged to refer to [5, 6] for a brief summary of several ongoing research activities at NIST regarding assembly related activities. The design for tolerance of electro-mechanical assemblies project [5, 7] advocates a more general and unified assembly representation scheme for proactive uses in the conceptual and detailed design phases. This scheme includes function, behavior and tolerance information models, along with other assembly information i.e., the geometric, topological and mating constraints, in the assembly data model. The primary goal of this project was to integrate comprehensive function, assembly (artifact) and behavior models. The FAB (**Function-Assembly-Behavior**) data model was developed to capture product development-related issues from the conceptual design stage to the detailed assembly building process. The proposed aggregate structure of function, behavior and assembly in this data model can support conceptual design as well as design for manufacturing and assembly, starting from an early design stage.

The primary objective of the integrated NIST Core Product Model (CPM) [8] is to provide a base-level product model that is: not tied to any vendor software; open; non-proprietary; simple; generic; expandable; independent of any one product development process; and capable of capturing the engineering context that is most commonly shared in product development activities. The core model focuses on artifact representation including function, form, behavior and material, physical and functional decompositions, and relationships among these concepts. The model is heavily influenced by the Entity-Relationship data model; accordingly, it consists of two sets of classes, called object and relationship, equivalent to the UML class and association class, respectively. It is expected that the core model may eventually serve as a precursor for STEP in the lifecycle of a product, capturing all information relevant to the ongoing design process until the product design is firmed up, approved and committed to purchasing or manufacturing.

The CPM provides several primitives, which we discuss next. The CPM focuses on artifact representation including function, form, behavior, and material, physical, and functional decompositions, and relationships among these concepts. An **Artifact** refers to a product or one of its components. (We use bold face notation for classes and packages). It is the aggregation of **Function**, **Form**, and **Behavior**. **Form** is the aggregation of **Geometry** and **Material**. In addition, an **Artifact** has **Specification** and **Feature**. The **Specification** refers to the general information that contains all the design requirements pertaining to the artifacts function or form. **Feature** represents any information in the **Artifact** that is an aggregation of **Function** and **Form**. For more information on the CPM, including the relationships (associations) defined between the classes shown; please refer to [8]. In the recent literature a lot of efforts have been made to represent function and form to high level of maturity [9].

In the ESPRIT funded project, known as MOKA [10], the product model supports five distinct views of the product: structure, function, behavior, technology, and representation. These views represent different perspectives of the underlying product model. The MOKA product representation model is similar to the FAB and CPM models and it includes a considerable amount of assembly information. However, the MOKA system does not represent kinematics, tolerance, and assembly and parametric constraints. The proposed model OAM can handle these types of constraints in addition to the constraints described in MOKA. These are essential for assembly, kinematics, and tolerance representations. The representations of the physical structure are supported within MOKA by the representation View, which includes geometry, and the finite element method (FEM). A separate class structure for FEM may be useful in design and analysis integration. However, it is included more as a place holder. A detailed description of the MOKA methodology for the development of knowledge based engineering applications is given in [11].

There are some academic systems that offer some facilities to represent assembly information. One such system developed by Whitney and Mantripragada [12] represents the high-level assembly information as the key characteristics. The chains of dimensional relationships and constraints in the product are handled by the so-called Datum Flow Chain concept [13]. One of the earlier works on assembly modeling was reported in [14]. The system of van der Net [15] focuses on designing assemblies taking into account requirements from the assembly process planning phase, in order to prevent design errors, reduce lead times, and be able to automate process planning. These requirements are captured in the assembly by specifying geometric, assembly and tolerance specific relations on and between the assembled parts. An excellent work on the integration of the views supporting parts design and assembly design of the whole product has been done by Noort et al. [16]. Callahan and Heisserman proposed a strategy for evaluating, comparing, and merging design alternatives [17]. Assembly features has also been subject to many studies [18–22].

3. UML Representation of the OAM

Most electromechanical products are assemblies of components. The aim of the Open Assembly Model (OAM) is to provide a standard representation and exchange protocol for assembly and system-level tolerance information. OAM is extensible; it currently provides for tolerance representation and propagation, representation of kinematics, and engineering analysis at the system level [23]. The assembly information model emphasizes the nature and information requirements for part features and assembly relationships. The model includes both assembly as a concept and assembly as a data structure. For the latter it uses the model data structures of STEP. OAM is an open model because it is independent from any implementation aspect; it is non-proprietary and can seamlessly interoperate with any application or generic analysis.

Figure 1 shows the main schema of the Open Assembly Model. The schema incorporates information about assembly relationships and component composition; the representation of the former is by the class **AssemblyAssociation**, and the model of the latter uses part-of relationships. The class **AssemblyAssociation** represents the component assembly relationship of an assembly. It is the aggregation of one or more **ArtifactAssociation**.

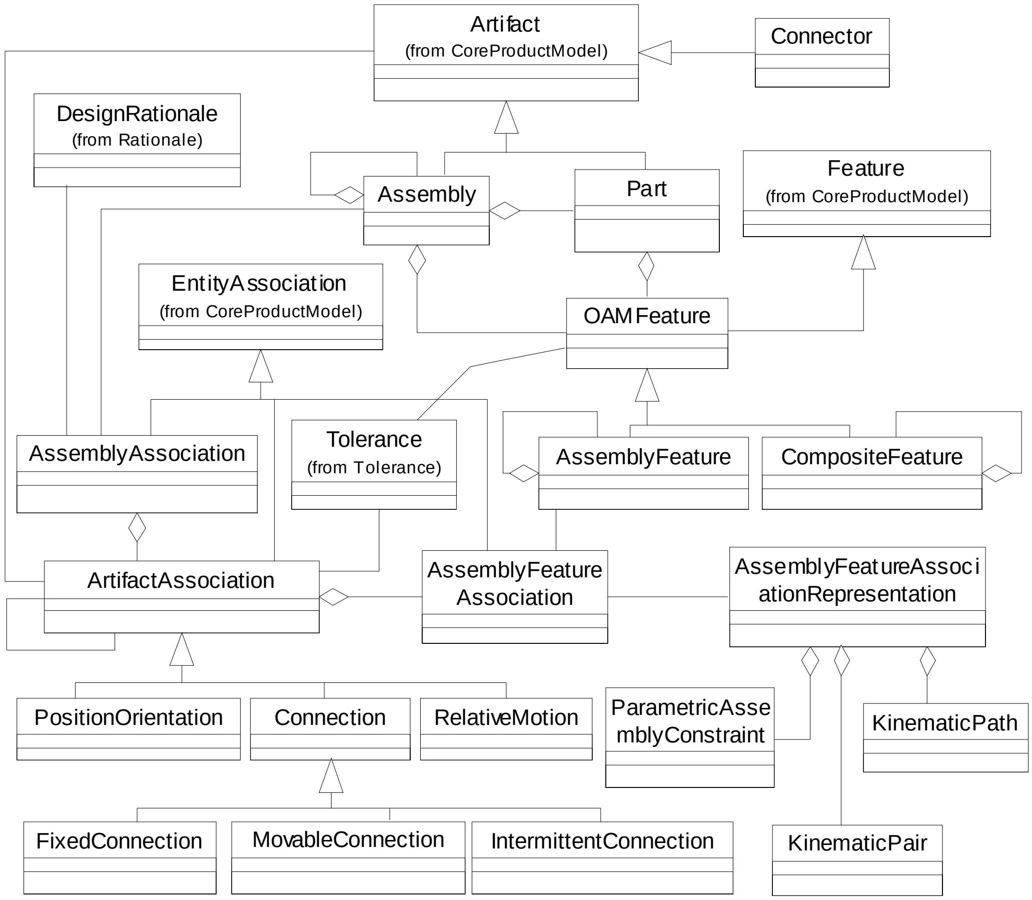

Figure 1: Class Diagram of the Open Assembly Model.

An **ArtifactAssociation** class represents the assembly relationship between one or more artifacts. For most cases, the relationship involves two or more artifacts. In some cases, however, it may involve only one artifact to represent a special situation. Such a case may occur when one fixes an artifact in space for anchoring the entire assembly with respect to the ground. It can also occur when we capture kinematic information between an artifact at an input point and the ground. We can regard such cases as relationships between the ground and an artifact. Hence, we allow the artifact association with one artifact associated in these special cases.

An **Assembly** is a composition of its subassemblies and parts. A **Part** is the lowest level component. Each assembly component (whether a sub-assembly or part) is made up of one or more features, represented in the model by **OAMFeature**. The **Assembly** and **Part** classes are sub-classes of the CPM **Artifact** class and **OAMFeature** is a subclass of the CPM **Feature** class. In CPM, **Geometry** and **Material** aggregate into **Form**. **Form** and **Function** aggregate into CPM **Feature** class. **ArtifactAssociation** is specialized into the following classes: **PositionOrientation**, **RelativeMotion** and **Connection**. **PositionOrientation** represents the relative position and orientation between two or more artifacts that are not physically connected and describes the associated constraints between the artifacts. **RelativeMotion** represents the relative motions between two or more artifacts that are not physically connected and describes the associated constraints between the artifacts. **Connection** represents the connection between artifacts that are physically connected.

Connection is further specialized as **FixedConnection**, **MovableConnection**, or **IntermittentConnection**. **FixedConnection** represents a connection in which the participating artifacts are physically connected and describes the type and/or properties of the fixed joints. **MovableConnection** represents the connection in which the participating artifacts are physically connected and movable with respect to one another and describes the type and/or properties of kinematic joints. **IntermittentConnection** represents the connection where the participating artifacts physically connect only intermittently (e.g. cam). **Connector** realizes **Connection**, which is a specialization of the **Artifact**.

OAMFeature has tolerance information, represented by the class **Tolerance**, and subclasses **AssemblyFeature** and **CompositeFeature**. **CompositeFeature** represents a composite feature that is decomposable into multiple simple features. **AssemblyFeature**, a sub-class of **OAMFeature**, by definition represents assembly features. Assembly features are a collection of geometric entities of artifacts. They may be partial shape elements of any artifact. For example, consider a shaft-bearing connection. The bearing's hole and a shaft's cylinder can be viewed as the assembly features that describe the physical connection between the bearing and the shaft. We can also think of geometric elements such as planes, spheres, cones, and tori as assembly features.

The class **AssemblyFeatureAssociation** represents the association between mating assembly features through which relevant artifacts are associated. The class **ArtifactAssociation** is the aggregation of **AssemblyFeatureAssociation**. Since associated artifacts can have multiple feature-level associations when assembled, one artifact association may have several assembly features associations at the same time. That is, an artifact association is the aggregation of assembly feature associations. Any assembly feature association relates in general to two or more assembly features. However, as in the special case where an artifact association involves only one artifact, it may involve only one assembly feature when the relevant artifact association has only one artifact. The class **AssemblyFeatureAssociationRepresentation** represents the assembly relationship between two or more assembly features. This class is an aggregation of parametric assembly constraints, a kinematic pair, and/or a relative motion between assembly features. **ParametricAssemblyConstraint** specifies explicit geometric constraints between artifacts of an assembled product, intended to control the position and orientation of artifacts in an assembly. Parametric assembly constraints are defined in ISO 10303-108 [24]. This class is further specialized into specific types: **Parallel**, **ParallelWithDimension**, **SurfaceDistanceWithDimension**, **AngleWithDimension**, **Perpendicular**, **Incidence**, **Coaxial**, **Tangent**, and **FixedComponent**.

KinematicPair defines the kinematic constraints between two adjacent artifacts (links) at a joint. The kinematic structure schema in ISO 10303-105 defines the kinematic structure of a mechanical product in terms of links, pairs, and joints [25]. The kinematic pair represents the geometric aspects of the kinematic constraints of motion between two assembled components. **KinematicPath** represents the relative motion between artifacts. The kinematic motion schema in ISO 10303-105 defines kinematic motion [25].

Tolerancing is a critical issue in the design of electro-mechanical assemblies. Tolerancing includes both tolerance analysis and tolerance synthesis. In the context of electro-mechanical assembly design, tolerance analysis refers to evaluating the effect of variations of individual part or sub-assembly dimensions on designated dimensions or functions of the resulting assembly. Tolerance synthesis refers to allocation of tolerances to individual parts or sub-assemblies based on tolerance or functional requirements on the assembly. Tolerance design is the process of deriving a description of geometric tolerance specifications for a product from a given set of desired properties of the product. Existing approaches to tolerance analysis and synthesis entail detailed knowledge of the geometry of the assemblies and are mostly applicable only during advanced stages of design, leading to a less than optimal design. In [26], a computational model for validating the dimensioning scheme and tolerance specifications compatible with dimensioning and tolerancing practice is presented.

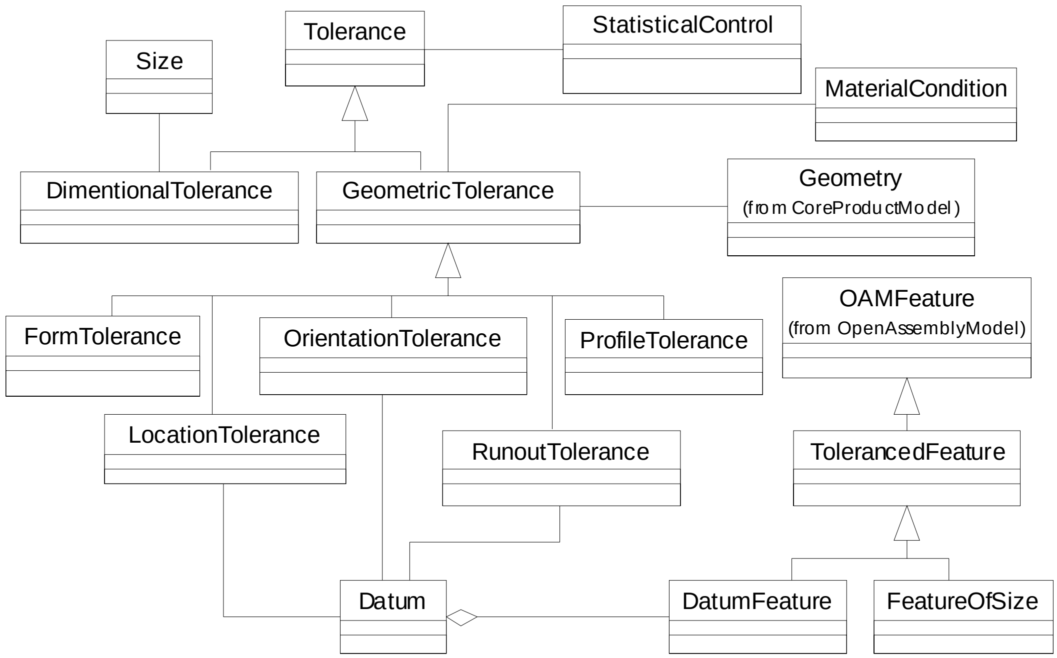

Figure 2: Tolerance Model.

During the design of an assembly, both the assembly structure and the associated tolerance information evolve continuously; we can achieve significant gains by effectively using this information to influence the design of that assembly. Any proactive approach to assembly or tolerance analysis in the early design stages will involve making decisions with incomplete information models. In order to carry out early tolerance synthesis and analysis in the early design stage, we include function, tolerance, and behavior information in the assembly model; this will allow analysis and synthesis of tolerances even with the incomplete data set. In order to achieve this we define a class structure for tolerance specification, and we show this in Figure 2.

DimensionalTolerance typically controls the variability of linear dimensions that describe location, size, and angle; it is also known as tolerancing of perfect form. This concept is included to accommodate the requirements of ISO 1101 standard [27]. **GeometricTolerance** is the general term applied to the category of tolerances used to control form, orientation, position, and runout. It enables tolerances to be placed on attributes of features, where a feature is one or more pieces of a part surface; feature attributes include size (for certain features), position (certain features), form (flatness, cylindricity, etc.), and relationship (e.g., perpendicular-to). The class **GeometricTolerance** is further specialized into the following: (1) **FormTolerance**; (2) **ProfileTolerance**; (3) **RunoutTolerance**; (4) **OrientationTolerance**; and (5) **LocationTolerance**.

Datum is a theoretically exact or a simulated piece of geometry, such as a point, line, or plane, which serves as a reference to a tolerance. **DatumFeature** is a physical feature that is applied to establish a datum. **FeatureOfSize** is a feature that is associated with a size dimension, such as the diameter of a spherical or cylindrical surface or the distance between two parallel planes. **StatisticalControl** is a specification that incorporates statistical process controls on the toleranced feature in manufacturing. A detailed description of tolerance model including a case study example will be given in a forthcoming paper.

4 Example and Industrial Case Study

This section illustrates the assembly model with an industrial device: a planetary gear system. The model is generated using a Computer Aided Design (CAD) system. Section 4.1 describes the principal hierarchy of the assembly. Section 4.2 explains the assembly relationships such as artifact associations, assembly feature associations, assembly constraints, and kinematic pairs.

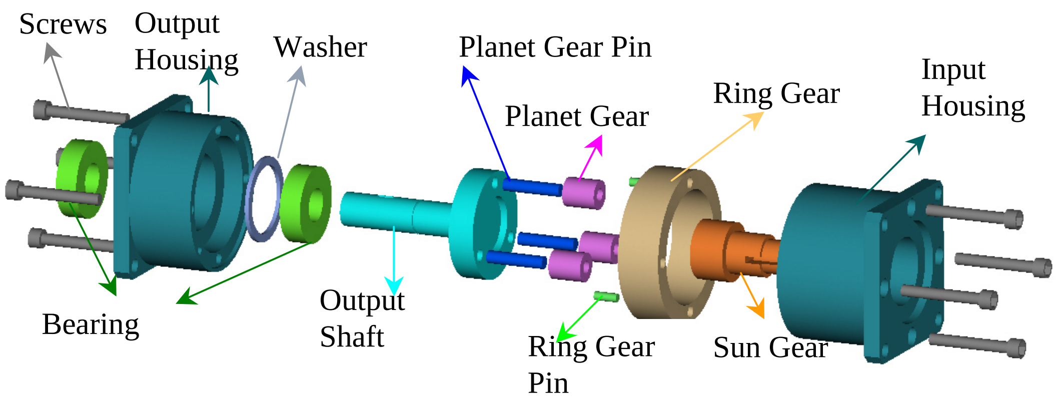

Figure 3: Exploded View of the Planetary Gear Model.

4.1 Assembly Hierarchy

A planetary gear system is used to illustrate our model. Before proceeding further with our case study example, we first need to define an assembly hierarchy for the planetary gear system. The planetary gear system is assumed to be composed of three parts, namely, the input-housing, the output-housing and the sungear, and five sub-assemblies, as shown in Figure 3. The five sub-assemblies are: (1) the output end assembly that contains the two bearings, the washer, and the output housing; (2) the ring gear assembly that consists of the ring gear and ring gear pin; (3) the planet carrier assembly that consists of three planet gear pins and the output shaft, and (4) the planet gear carrier assembly that consists of the three planet gears and the planet carrier assembly, and (5) the planet gear carrier and sungear assembly.

The details of the assembly relationships are explained only for some of the artifacts to avoid repetition. The assembly of the ring gear and the planet gear carrier subassemblies with the output housing and the input housing is not shown in this paper to avoid repetition, interested readers can see [28] for more details.

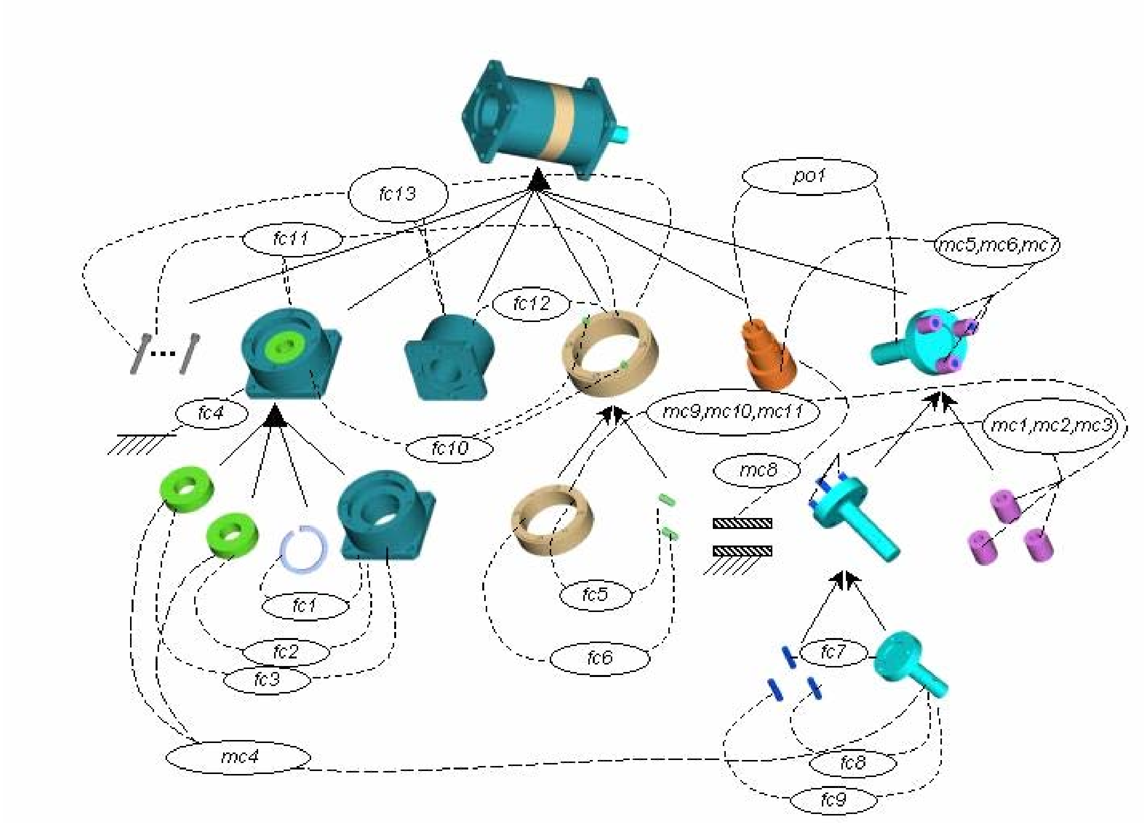

Figure 4: Artifact Associations in the Planetary Gear System.

Notice that the hierarchy in Figure 4 is introduced to verify and demonstrate the proposed UML assembly model. The root node is the entire assembly, the interior nodes are sub-assemblies, and the leaf nodes are component parts. The sequence of part assembly descriptions does not imply the actual assembly sequence. The assembly sequencing task is outside the scope of this paper.

4.2 Assembly Relationships

Information besides the hierarchical relationship between artifacts is provided by an instance of **AssemblyAssociation**, which is an aggregation of instances of **ArtifactAssociation**. The artifact associations hold relational information such as mating conditions, kinematic pairs, and associations between assembly features. A graph of artifact associations is shown in Figure 4. The dotted lines indicate the artifact associations and the solid lines portray the hierarchical assembly relationships. The details of the artifact associations shown in Figure 4 are discussed in the following sections. We will explain the assembly relationship for subassemblies other than output and input housing subassemblies.

4.2.1 Ring Gear Assembly

The ring gear subassembly is shown in Figure 5. It consists of three parts: ring gear, ring gear pin 1 and pin 2. The two ring gear pins go into the pinholes of the ring gear with a tight fit.

The assembly relationships are listed in Table 1, and Figure 6 shows the instance diagram of the current assembly. The instance names take the form of “instance name:class name”. The artifact associations are instantiated from **FixedConnection** and named *fc5* and *fc6*, since there are no relative motions between participating artifacts. The artifact association *fc6* has a similar structure to that of *fc5* and is not shown in the figure.

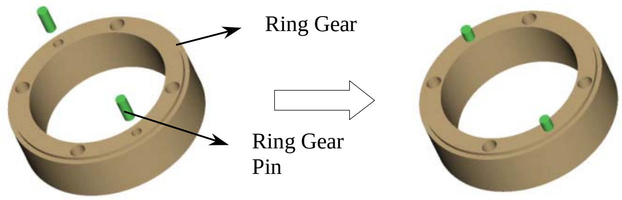

Figure 5: Ring Gear Assembly.

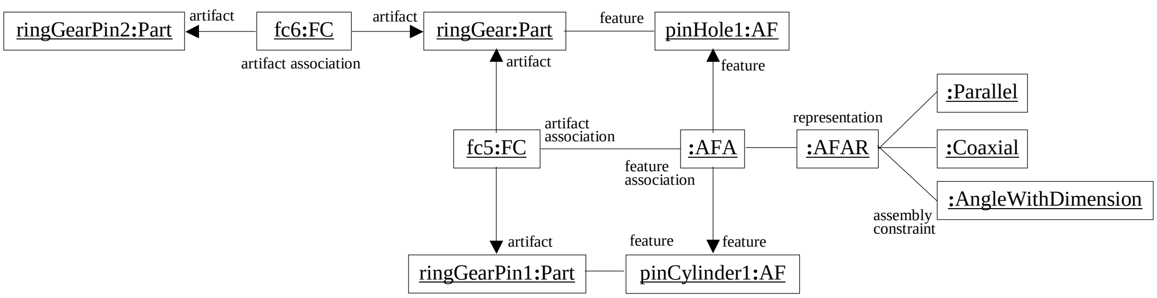

Figure 6: Instance Diagram of Ring Gear Assembly.

Artifact assoc.	Artifacts	Assembly features	Assembly constraints	Kinematic relationships
fc5	Ring gear	Pinhole surfaces (pinHole1: AF)	Coaxial Parallel Angle with dimension	No relative motion
	Ring-gear pin 1	Inserted portion of pin surface (pinCylinder1:AF)		
fc6	Ring gear	Pinhole surfaces (pinHole2:AF)	Coaxial Parallel Angle with dimension	No relative motion
	Ring-gear pin 2	Inserted portion of pin surface (pinCylinder2:AF)		

Table 1: Assembly Relationships of Ring Gear Assembly.

4.2.2 Planet Carrier Assembly

The planet carrier assembly in Figure 7 is comprised of four parts: three planet gear pins and an n output shaft. The three planet gear pins are assembled with output shaft by a tight fit.

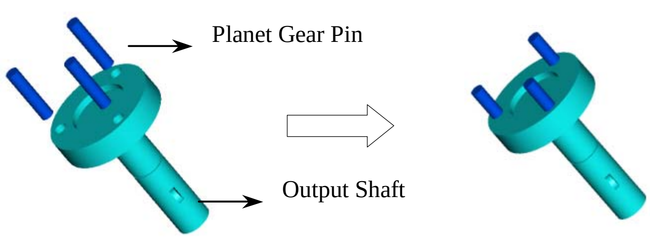

Figure 7: Planet Carrier Assembly.

The assembly relationships are listed in Table 2, and the instance diagram is depicted in Figure 8.The artifact associations are instantiated from **FixedConnection** and named *fc7*, *fc8*, and *fc9*, since there are no relative motions between participating artifacts. The assembly relationships of the current assembly are very similar to those of the ring gear assembly explained previously. The detailed relationships for *fc8* and *fc9* are not shown in the figure: they have the same structure as that of *fc7*.

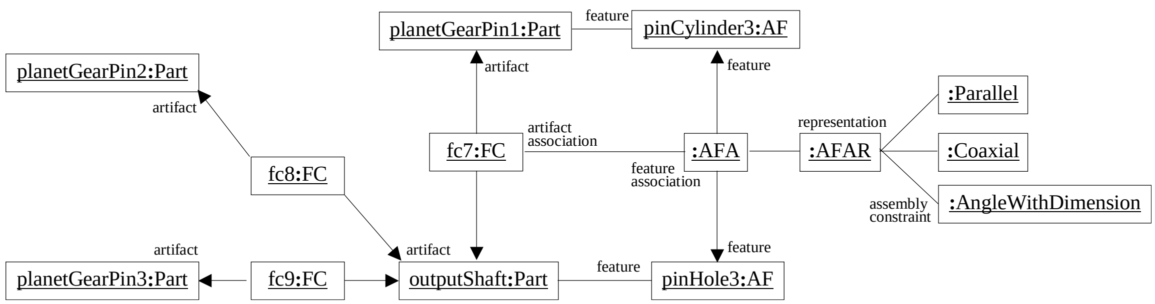

Figure 8: Instance Diagram of Planet Carrier Assembly.

Artifact assoc.	Artifacts	Assembly features	Assembly constraints	Kinematic relationships
fc7	Output shaft	Pinhole surfaces (pinHole3:AF)	Coaxial Parallel Angle with dimension	No relative motion
	Planet-gear pin 1	Inserted portion of pin surface (pinCylinder3:AF)		
fc8	Output shaft	Pinhole surfaces (pinHole4:AF)	Coaxial Parallel Angle with dimension	No relative motion
	Planet-gear pin 2	Inserted portion of pin surface (pinCylinder4:AF)		
fc9	Output shaft	Pinhole surfaces (pinHole5:AF)	Coaxial Parallel Angle with dimension	No relative motion
	Planet-gear pin 3	Inserted portion of pin surface (pinCylinder5:AF)		

Table 2: Assembly Relationships of Planet Carrier Assembly.

4.2.3 Planet Gear Carrier Assembly

The planet gear carrier assembly shown in Figure 9 is comprised of four artifacts: three parts of planet gears and the planet carrier assembly. The three planet gears are assembled by loose fit with the planet gear pins of the planet carrier assembly.

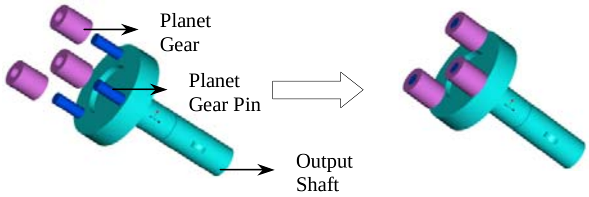

Figure 9: Planet Gear Carrier Assembly.

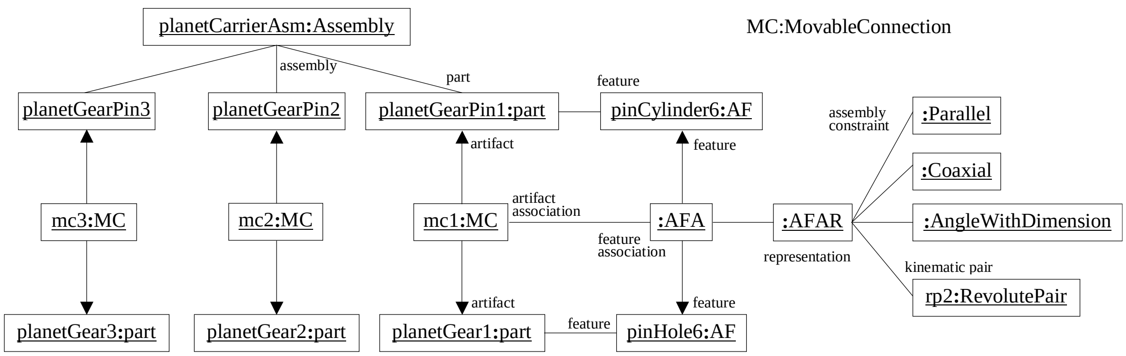

Figure 10: Instance Diagram of Planet Gear Carrier Assembly.

Table 3 lists the assembly relationships. The instance diagram of these assembly relationships is illustrated in Figure 10. The artifact associations of the current assembly are instantiated from **MovableConnection** since there are rotational motions between the planet gears and the planet gear pins. They are named *mc1*, *mc2*, and *mc3*, respectively. Only the details of artifact association *mc1* are depicted. The instance of the planet carrier assembly (planetCarrierAsm:Assembly) is also drawn to show the part-of relationships with the planet gear pins; the output shaft is not shown since it is not directly involved in the current assembly relationship. Note that the part-of relationships are actually stored in the main hierarchy of the proposed UML model. As mentioned above, the artifact associations are instances of **MovableConnection**. Thus, the associated assembly feature association representations contain the information on the kinematic pair. Instances of **RevolutePair** are thus supplied to the assembly feature association representation, as well as the assembly constraints.

Artifact assoc.	Artifacts	Assembly features	Assembly constraints	Kinematic relationships
mc1	Planet-gear pin 1 @Planet carrier assembly	Pin surface for planetary gear (pinCylinder6:AF)	Coaxial Parallel Angle with dimension	Relative rotation (rp2:RevolutePair)
	Planet gear 1	Gear journal surface (pinHole6:AF)		
mc2	Planet-gear pin 2 @Planet carrier assembly	Pin surface for planetary gear (pinCylinder7:AF)	Coaxial Parallel Angle with dimension	Relative rotation (rp3:RevolutePair)
	Planet gear 2	Gear journal surface (pinHole7:AF)		
mc3	Planet-gear pin 3 @Planet carrier assembly	Pin surface for planetary gear (pinCylinder8: AF)	Coaxial Parallel Angle with dimension	Relative rotation (rp4:RevolutePair)
	Planet gear 3	Gear journal surface (pinHole8:AF)		

Table 3: Assembly Relationships of Planet Gear Carrier Assembly.

4.2.4 Planet Gear Carrier and Sungear Assembly

The sungear is assembled to the planet gear carrier subassembly with the three planet gears by gear meshing (Figure 11). The assembly relationships are listed in Table 4 and the instance diagrams are illustrated in Figure 12. Five parts participate in the current assembly relationships. Three movable connections *mc5*, *mc6,* and *mc7* are instantiates of **MovableConnection**.

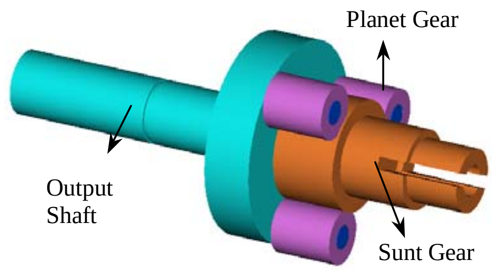

Figure 11: Planet Gear Carrier and Sungear Assembly.

Artifact assoc.	Artifacts	Assembly features	Assembly constraints	Kinematic relationships
mc5	Planet-gear 1 @Planet gear-carrier assembly	Gear teeth surface (teeth7:AF)	None	Gear meshing (gp1:GearPair)
	Sungear	Gear teeth surface (teeth1:AF)		
mc6	Planet-gear 2 @Planet gear-carrier assembly	Gear teeth surface (teeth9:AF)	None	Gear meshing (gp2:GearPair)
	Sungear	Gear teeth surface (teeth2:AF)		
mc7	Planet-gear 3 @Planet gear-carrier assembly	Gear teeth surface (teeth11:AF)	None	Gear meshing (gp3:GearPair)
	Sungear	Gear teeth surface (teeth3:AF)		
po1	Output shaft @Planet carrier assembly @Planet gear-carrier assembly	Whole part (outputShaftFeature:AF)	Coaxial Parallel Angle with dimension	N/A
	Sungear	Whole part (sunGearFeature: AF)		
mc8	Sungear	Input shaft surface (inputShaft:AF)	None	Relative rotation (rp1:RevoluteP air)

Table 4: Assembly Relationships between Planet Gear Carrier Assembly and Sungear.

To describe the details of the gear meshing, three instances of **GearPair**, namely, *gp1*, *gp2*, and *gp3* in Figure 12 are attached to the respective artifact associations (movable connections) via matching assembly feature associations. On the other hand, the input shaft portion of the sungear has relative rotation with respect to an unknown support (or ground). Typically, it is coupled with the output shaft of a motor. The output shaft of the motor would have relative rotation with respect to the support (or ground). That is, there is only one artifact (part) involved in this kinematic relationship. To handle this case, we may use an artifact association with one artifact participating, as the instance *mc8* described in Table 4 and Figure 12. Its associated assembly feature association also has only one assembly feature. The kinematic relationship is captured by an instance rp1, which is an instance of **RevolutePair** and attached to the assembly feature association as shown in Figure 12. On the other hand, to position the sungear, parametric assembly constrains need to be assigned.

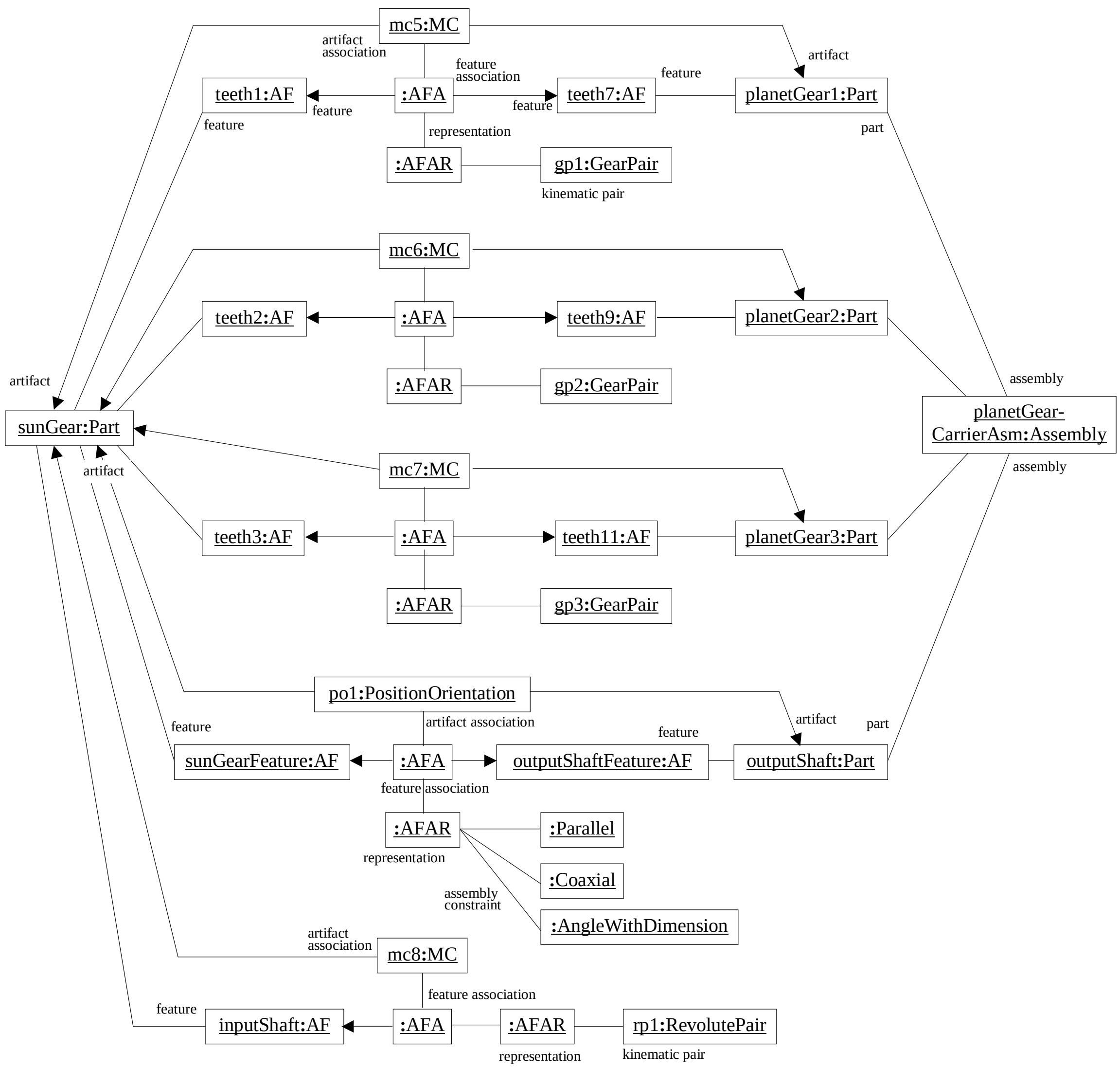

Figure 12: Instance Diagram of Planet Gear Carrier and Sungear Assembly.

In this example, it is assumed that the sungear is positioned with respect to the output shaft. Since they are not directly connected, the classes specialized from **Connection**, which are used for artifacts physically connected, cannot be used to represent this relationship. Instead, the relative position and orientation between two artifacts that are not physically connected can be captured using the **PositionOrientation** class which is specialized from **ArtifactAssociation** (see *po1* in Figure 12).

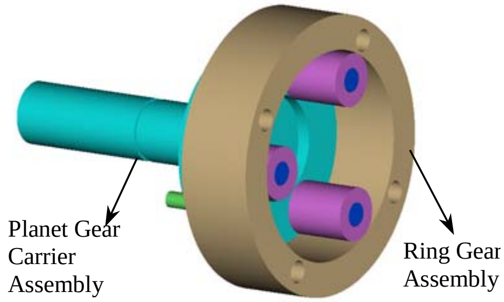

Figure 13: Planet Gear Carrier and Ring Gear Assembly.

The two artifacts in the above case do not have a direct contact, and thus the mating assembly features cannot be identified. This situation, however, may be handled by using assembly features representing the whole artifact, or dummy (null) features. In this example, we assume that the artifacts, as a whole, are the involved assembly features. They are named sunGearFeature:AF and outputShaftFeature:AF. An instance of assembly feature association representation incorporating the necessary parametric assembly constraints is shown in Figure 12.

4.2.5 Planet Gear Carrier and Ring Gear Assembly

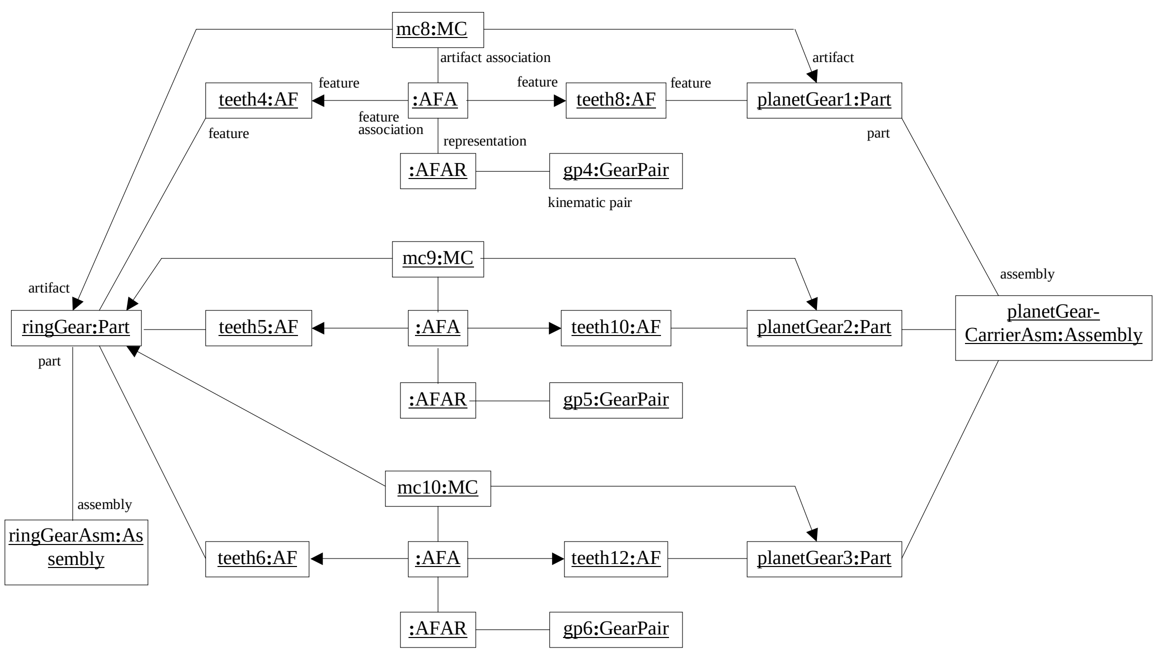

Figure 14: Instance Diagram of Planet Gear Carrier and Ring Gear Assembly.

Artifact assoc.	Artifacts	Assembly features	Assembly constraints	Kinematic relationships
mc9	Planet-gear 1 @Planet gear-carrier assembly	Gear teeth surface (teeth8:AF)	None	Gear meshing (gp4:GearPair)
	Ring gear @Ring gear assembly	Gear teeth surface (teeth4:AF)		
mc10	Planet-gear 2 @Planet gear-carrier assembly	Gear teeth surface (teeth10:AF)	None	Gear meshing (gp5:GearPair)
	Ring gear @Ring gear assembly	Gear teeth surface (teeth5:AF)		
mc11	Planet-gear 3 @Planet gear-carrier assembly	Gear teeth surface (teeth12:AF)	None	Gear meshing (gp6:GearPair)
	Ring gear @Ring gear assembly	Gear teeth surface (teeth6:AF)		

Table 5: Assembly Relationships between Ring Gear Assembly and Planet Gear-Carrier Assembly.

Let us now consider the assembly of planet gear carrier and ring gear shown in Figure 13 (note that this figure does not show the sungear which is already assembled to the planet gear carrier. The sungear does not participate in the assembly described in this section). The ring gear is meshed with the three planet gears of the planet gear carrier assembly. The assembly relationships are shown in Table 5, and Figure 14 illustrates the instance diagram of the assembly. The artifact associations are very similar with those of the previous sungear and the planet gear carrier assembly. As in the previous example, three artifact associations *mc9*, *mc10*, and *mc11* are instances of **MovableConnection**, and *gp4*, *gp5*, and *gp6* are instances of **GearPair**.

5 Conclusions and Future Work

In this paper, we described an object-oriented UML representation of an assembly model for electro-mechanical products representation. This model incorporates tolerance representation, kinematics, assembly relationships, and assembly features. The Open Assembly Model (OAM) described in this paper is based on the class structure of the NIST Core Product Model [8]. The classes defined in OAM, for example **Assembly**, inherit function, behavior, and form from the Core Product Model's **Artifact** class. The UML model of the assembly is described with an example. Tolerance and kinematics analyses of this system are used to show how such an assembly model can be exploited by designers. We are planning to populate this model further and make it interoperate with various CAD and engineering analysis systems. Further we will explore the possibilities of integrating it with virtual reality systems such as VADE [29].

Acknowledgements

The authors wish to acknowledge the valuable comments and improvements suggested by Prof. Steven Fenves. Those comments have substantially improved and shaped the paper. This project is funded [in part] by NIST's Systems Integration for Manufacturing Applications (SIMA) Program. SIMA supports NIST projects applying information technologies and standards-based approaches to manufacturing software integration problems

Disclaimer

No approval or endorsement of any commercial product by the National Institute of Standards and Technology is intended or implied. Certain commercial equipments, instruments, or materials are identified in this paper in order to facilitate better understanding. Such identification does not imply recommendations or endorsement by the National Institute of Standards and Technology, nor does it imply the materials or equipment identified are necessarily the best available for the purpose.

References

[1] Booch, G., Rumbauch, J., and Jacobson, I., 1997, "The Unified Modeling Language User Guide", Addison-Wesley.

[2] ISO, 1994, “STEP ISO 10303 - Industrial automation systems and integration - Product data representation and exchange - Part 44: Integrated Resources: Product Structure Configuration”, ISO, Geneva, CH.

[3] Sugimura, N., 2002, “ISO/CD 10303-109, Product data representation and exchange: Integrated application resource: Kinematic and geometric constraints for assembly models”, ISO, Geneva, CH.

[4] Shah, J. J., and Mantyla, M, 1995, “Parametric and Feature-Based CAD/CAM: Concepts, Techniques, and Applications”, Wiley-Inderscience, October 1995.

[5] Narahari, Y., Sudarsan, R., Lyons, K. W., Duffey, M. R., and Sriram, R. D., 1999, “Design for Tolerance of Electro-Mechanical Assemblies: An Integrated Approach”, IEEE Trans. Robotics and Automation, 15(6):1062-1079.

[6] Sudarsan, R., Roy, U., Narahari, Y., Sriram R. D., Lyons, K.W., and N. Pramanik N., 2000, “Information Models for Design Tolerancing: From Conceptual to the Detail Design”, Technical Report NISTIR 6524, National Institute of Standards and Technology, Gaithersburg, MD, USA.

[7] Sriram, R. D., 1999, “Standards for the Collaborative Design Enterprise-Response to OMG's MfgDTF RFI#4”, http://cgi.omg.org/docs/mfg/99-08-04.pdf.

[8] Fenves, J. S., 2001, “A Core Product Model for Representing Design Information”, Technical Report NISTIR 6736, National Institute of Standards and Technology, Gaithersburg, MD, USA.

[9] Chang, E., Li, X., and Schmidt, L. C., 2000, “The need for form, function, and behavior-based representations in design”, Technical report, DATLab, University of Maryland, USA.

[10] MOKA. “MOKA: A Framework for structuring and representing engineering knowledge”, http://www.kbe.coventry.ac.uk/moka/miginfo.htm
.

[11] Stokes, M., editor, 2001, Managing Engineering Knowledge: MOKA Methodology for Knowledge Based Engineering Applications. ASME.

[12] Whitney, D. E., and Mantripragada, R., 1998, “The Datum Flow Chain: A systematic approach to assembly design and modeling”, In ASME Design Engineering Technical Conferences and Computers in Engineering Conference. ASME.

[13] Whitney, D. E., 2004, Mechanical Assemblies: Their Design, Manufacture, and Role in Product Development. Oxford University Press.

[14] Lee, K., and Gossard, D. C., 1985, “A hierarchical data structure for representing assemblies: Part 1”, CAD, 17(1):15-19.

[15] Van der Net, A., 1998, “Designing and manufacturing assemblies”, PhD thesis, Eindhoven University of Technology.

[16] Noort, A., Hoek, G. F. M., and Bronsvoort, W. F., 2002, “Integrating part and assembly modeling”, CAD, 34(12):899-912.

[17] Callahan, S., and Heisserman, J., 1997, “A product representation to support process automation”, In M.J. Pratt, R. D. Sriram, and M.J. Wozny, editors, Product Modeling for Computer Integrated Design and Manufacture, pages 285-296. Chapman and Hall, 1997.

[18] Shah, J. J., 1991, “Assessment of features technology”, CAD, 23(5):331-343.

[19] Shah, J. J., and Rogers, M. T., 1993, “Assembly modeling as an extension of feature-based design”, Research in Engineering Design, 5:218-237.

[20] Van Holland, W., and Bronsvoort W. F., 2000, “Assembly features in modeling and planning”, Robotics and Computer Integrated Manufacturing, 16(4):277-294.

[21] Coma, O., Mascle, C., and Véron, P., 2003, “Geometric and form feature recognition tools applied to a design for assembly methodology”, CAD, 35(13):1193-1210.

[22] Chan, C. K., and Tan, S. T, 2003, “Generating assembly features onto split solid models”, CAD, 35(14):1315-1336.

[23] ISO, 1994, “ISO 10303-1 Industrial Automation Systems and Integration - Product Data Representation and Exchange - Part 1: Overview and fundamental principles”, ISO, Geneva, CH.

[24] ISO, 2003, “ISO 10303-108. Product data representation and exchange: Integrated application resource: Parameterization and constraints for explicit geometric product models”, ISO, Geneva, CH.

[25] ISO, 1996, “ISO 10303-105. Industrial automation systems and integration - Product data representation and exchange - Part 105: Integrated application resource: Kinematics”, ISO, Geneva, CH.

[26] Kandikjan, T., Shah, J. J., and Davidson, J. K., 2001, “A mechanism for validating dimensioning and tolerancing schemes in CAD systems”, CAD, 33(10):721-737.

[27] Kemmerer, S., editor, 1999, STEP: The Grand Experience, NIST Special Publication 939, National Institute of Standards and Technology, Gaithersburg, MD, USA.

[28] Sudarsan, R., Han, Y. H., Feng, S. C., Roy, U., Wang, F., Sriram, R. D., and Lyons, K. W, 2003, “Object-oriented representation of electro-mechanical assemblies using UML”, Technical Report NISTIR 7057, National Institute of Standards and Technology, Gaithersburg, MD, USA.

[29] Connacher, H., Jayaram, S., and Lyons K. W, 1997, “Virtual assembly using virtual reality techniques”, CAD, 29(8):575-584.