NISTIR 7057

Object-Oriented Representation of Electro-Mechanical Assemblies Using UML

Sudarsan Rachuri
Young-Hyun Han
Shaw C Feng
Utpal Roy
Fujun Wang
Ram D Sriram
Kevin W Lyons

NIST
National Institute of Standards and Technology
Technology Administration, U.S. Department of Commerce

NISTIR 7057

Object-Oriented Representation of Electro-Mechanical Assemblies Using UML

Sudarsan Rachuri
Young-Hyun Han
Shaw C Feng
Utpal Roy
Fujun Wang
Ram D Sriram
Kevin W Lyons

October 03

U.S. DEPARTMENT OF COMMERCE
Donald L. Evans, Secretary
TECHNOLOGY ADMINISTRATION
Phillip J. Bond, Under Secretary of Commerce for Technology
NATIONAL INSTITUTE OF STANDARDS AND TECHNOLOGY
Arden L. Bement, Jr., Director

Table of Contents

List of Figures

List of Tables

List of Acronyms

ASME	American Society of Mechanical Engineers
BOM	Bill of Materials
CAD	Computer Aided Design
CAM	Computer Aided Manufacturing
CPM	Core Product Model
DAG	Directed Acyclic Graph
DFT	Design For Tolerancing
ESPRIT	European Strategic Program on Research in Information Technology
FAB	Function Assembly Behavior
FEM	Finite Element Method
ISO	International Organization for Standardization
MOKA	Methodology and tools Oriented to Knowledge based engineering Applications
NIST	National Institute of Standards and Technology
OAM	Open Assembly Model
OpenADE	Open Assembly Design Environment
SIMA	System Integration for Manufacturing Applications
STEP	STandards for the Exchange of Product model data
UML	Unified Modeling Language
VADE	Virtual Assembly Design Environment

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 report 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 represent 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 model captures 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.

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. Hence, companies are increasingly staffing only their core competencies in-house and are depending on other firms to provide the complementary design knowledge and design effort needed for a complete product. 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, design 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. Although 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 part and assembly information between heterogeneous modeling systems is critical for unrestricted exchange of product data. However, little has been done in terms of developing standard representations that specify design information and product 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. Furthermore, we need to address the evolution of their corresponding information models during the conceptual and detailed design stages. In this report, we propose an integrated information model for assembly representation that is important for the exchange of information between modeling, analysis and planning systems.

The report is organized as follows. In Section 2, we discuss the existing ISO Standard for product data representation and exchange, ISO TC 184/SC4 [1], the Japanese National Committee (JNC) proposal for a STEP assembly model of products [2] and previous NIST efforts and current works towards assembly representations [3], [4], [5-7]. In Section 3, we present the object-oriented representation of electro-mechanical assemblies using UML [8]. We use bold face notation for classes. The various class definitions including, assembly, feature, kinematics, and tolerance are discussed. In Section 4, we present a Usecase analysis of the assembly model with a planetary gear system and explain how to use OAM in kinematic and tolerance representations. Object diagrams explaining the relationships, as well as the geometric and related data are also provided. Finally, conclusions and recommendations for further research work are presented in Section 5.

2 Previous Work

In this section we discuss the various research efforts toward developing assembly models, product representations, and standards. In Section 2.1, we describe the assembly representation in the ISO 10303 [9] standards. In Section 2.2, we discuss the recent proposal by an ISO working group towards enhancing the assembly representation in ISO 10303. In Section 2.3, we describe the relevant research projects at NIST, including Design for Tolerancing and the Core Product Model. We also present a brief discussion of several other assembly and product representation models developed by other researchers.

2.1 ISO Standard for Product Data Representation

ISO 10303-Part 44 [10] provides some limited assembly design representations that capture assembly structure and kinematic joint information during the design process. The assembly model establishes a neutral representation of assemblies of products, which are composed of sets of components. In this model, complete products are called "assemblies," and the components of 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. The application fields of interest to the assembly model are the kinematic analysis of assemblies, the animation of assemblies for digital-mockup technologies, assembly/disassembly process planning from the viewpoint of CAD and CAM systems, and tolerance analysis and synthesis. The product (assembly) data representation is divided into three major schemas:

The **Product structure schema** defines a product in terms of a nested decomposition into its constituents. It also defines mechanisms for expressing the composition relationship. Product structures are modeled by directed acyclic graphs (DAG). Nodes represent product definitions (PD) and arcs represent relationships (PDRs). Refer to

Figure 1. Product definition usage (PDU) refers to part/whole or composed-of relationships. It is a subtype of PDR and defines two subtypes:

- Assembly component usage (ACU) which establishes *composed-of* relationships between PDs.

- Make from usage option – (MFUO), which represents the fact that any actual unit of one design can be manufactured by using or modifying an actual unit of another design.

The entity ACU defines the following subtypes:

- quantified_ACU (QACU), which defines the relationship between component and assembly .

- next_CUOccurence (NCUO), which defines the immediate parent/child relationship.

- specified_higher_usage_occurrence (SHUO), which defines the specified parent/child relationship.

- promissory_usage_occurrence (PUO), which represents the relationship between a constituent and an ancestor assembly within an overall product structure without any specification of the intermediate assemblies being represented. The word "occurrence" is used in a sense to mean appearance/existence. The red lines in Figure 1 refer to the proposed ISO Part 109.

.

This product structure schema supports at least two forms of product structure:

1. Bill-of-Materials (BOM), and

2. Parts List Structure

The BOM data structure is used to represent the assembly aspects of a product structure. It includes only one instance of a single product definition for each kind of product that participates in the assembly structure. A part list structure is a specific form of a bill-of-material that can be represented by a tree (guided by the BOM's DAG structure). A bill-of-material structure may require a more general DAG. The parts list data structure individualizes the relationship between lower-level parts of the product structure with higher-level assemblies in which they are contained.

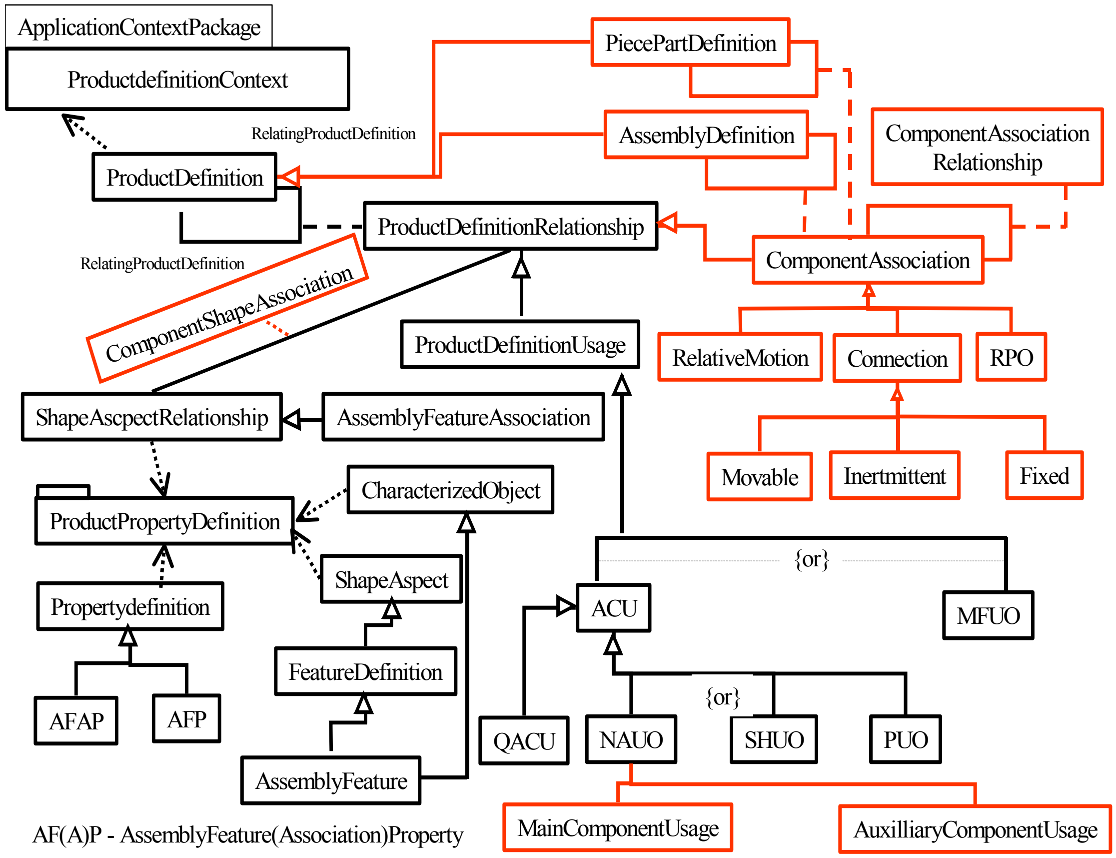


Figure 1: ISO 10303 Part Structure in UML Notation

Product concept schema describes a product as a set of specifications derived from customer needs analysis. It is limited to specification description only; it does not represent the product evolution process from the conceptual to the detailed design phase.

Configuration management schema manages the structure of the configuration of assemblies and components.

In summary, a primary feature of the entities defined in this part of ISO 10303 is that it provides the data modeler with various types of product structure data structures (e.g., BOM, parts list, etc.,) using the primitive entities. This facilitates meaningful information exchange between users without resorting to costly data model mapping tasks. However, ISO 10303 does not adequately address the following:

1) The relationship among different product definitions for the same product. For example, 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.

3) The decisions made and their rationale, throughout the entire product life cycle.

4) The physical connections among components of a product.

5) The properties that a product constituent may have.

6) The information for as-built manufacturing, manufacturing planning, and logistical structure and configurations.

7) Multiple versions of a single product that are not form (the shape of the product), fit (the fit is the way the product interfaces with other products), and function (the purpose that the product serves) equivalent.

2.2 ISO Working Group Proposal

The ISO working group (TC 184/SC4/WG12) *(*JNC proposal [2]*)* has proposed[1] several enhancements to STEP's assembly representation. In the WG12 proposal, the detailed geometric information not only for hierarchical relationships but for peer to peer relationships among component parts via an assembly feature is introduced. Geometric constraints among component parts at the detailed geometric element level are also enabled. The WG12 proposal introduces more information on component association and includes detailed information about appropriate assembly features involved in component association.[2]

Following ISO definitions, an assembly has been defined as a set of components and is represented in a similar hierarchical fashion. The assembly model includes:

1) *Information about individual parts*, which are piece parts defined by users and used in developing the assembly tree as required.

2) *Information about standard parts*, which refers to off-the-shelf standard parts such as nuts and bolts, keys, electric motors, etc. Standard parts can be of two

[1] The scope of this proposal has been modified to address the kinematic and geometric constraints for assembly models and the current status of this proposal is explained in [20].
[2] **Assembly feature** is defined as an element to specify the relationships between a pair of assembled components. For example, a hole and a cylinder are typical assembly features, which represent the physical connections between a bearing part and a shaft part.

types: (i) standard parts as discussed in other ISO standard parts catalogues (Reference: ISO TC 184/SC4/WG2), and (ii) those defined by the users. Note that the parts are all "solid" parts and non-solid parts are not considered.

3) *Product structure configuration*, which represents the hierarchical relationship between the components (assemblies, sub-assemblies and piece parts). Nominal and tolerance values of position and orientation of every component (a part, subassembly, or an assembly) are explicitly defined in its "parent" component (subassembly or assembly).

4) *Component association*, which provides detailed information for component association (i) between every pair of physically connected components; (ii) between every pair of components that is not physically connected; and (iii) detailed information about assembly features that are needed to define technological information of component association.

    a. *For a pair of physically connected components*: depending on the type of the associations, it can be of the three subtypes:

        i. *Fixed Connection*, such as a rigid joint. The pair is physically connected and fixed.
        ii. *Movable Connection*, which describes the association when the pair is physically connected and movable. Shaft-bearing joints, slider-guideway joints, gear joints, etc. are examples. Kinematic joint definition is applicable for its representation. It describes the possible relative motions between a pair of components and the properties of the joints, which constrains the components.
        iii. *Intermittent Connection*, which describes the association when the pair is physically connected with each other with only intermittent contacts. Examples are limit switches, cam-follower, etc.

    b. *For a pair of components that are not physically connected:*

        i. *Relative motion* describes the constraints on the relative motion between a pair of components.
        ii. *Relative position and orientation* describes the constraints on the relative position and orientation of a component against another component.
        iii. *Tolerance of the relative motions, positions, and orientations* represents associated tolerances for relative motions, positions, and orientations.

5) *Assembly Features*, which represent a region of a component, that is of interest in the assembly context. Assembly features may have (i) a shape-aspect (super class) as defined in 10303, and (ii) an assembly-feature-association. The assembly

feature associations and the assembly features describe the details of the component's associations.

The basic structure of the working group's proposed assembly model is shown in Figure 2. In the figure, boxes describe the components (assemblies, sub-assemblies, and parts), open circles the hierarchical relationships among the component definitions, and filled circles the associations between components. Note in the figure that the component association between sub-assembly3 and sub-assembly4 could be derived from the specified component association between part4 and part1, and hence may be redundant.

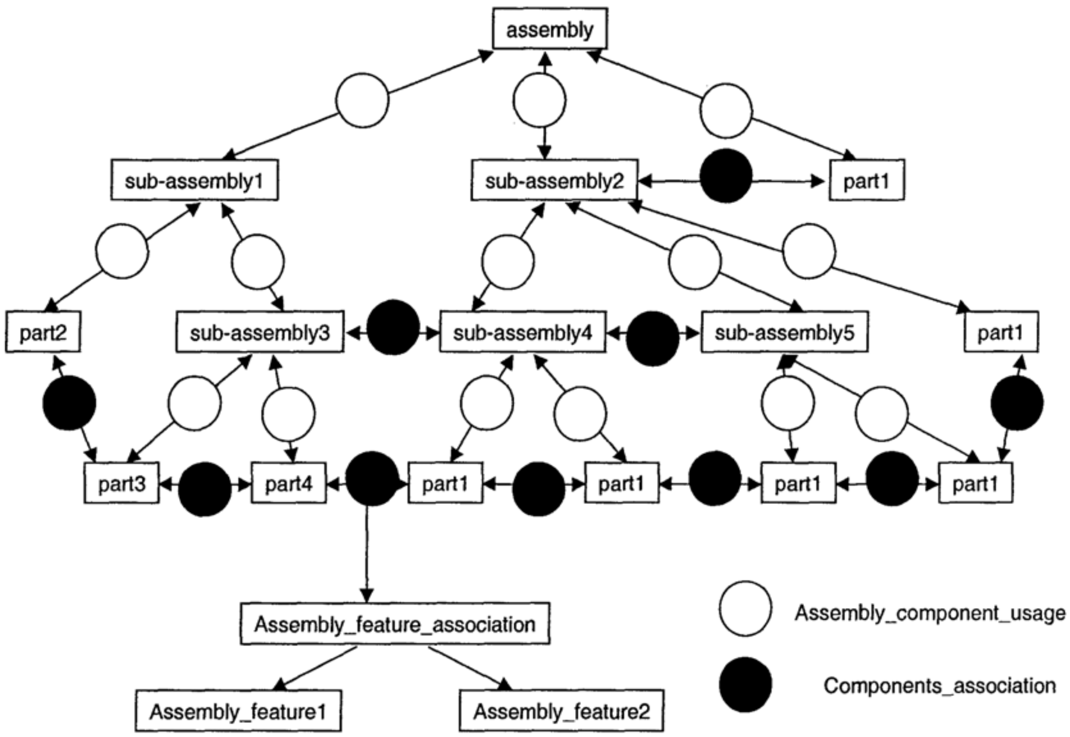


Figure 2: Assembly Model Proposed by JNC. Source [2].

It should be noted that the JNC proposal does not cover configuration management of the assemblies and components. Although the proposal outlines the possible applications of the proposed assembly representation in four fields - kinematic analysis of assemblies, animation of assemblies, assembly/disassembly process planning, and tolerance analysis and synthesis - actual application methodologies are not identified or reported.

2.3 Research at NIST

NIST has been actively involved in identifying and developing representational methodologies for the next generation of assembly related standards. The Open

Assembly Design Environment (OpenADE) [7] project 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 reference [11] for a brief summary of several ongoing research activities at NIST regarding assembly related activities. In the following sections, we discuss several projects relevant to the proposed OAM.

2.3.1 Open Assembly Design Environment Project

The OpenADE project is an initiative at NIST to provide an integrated and augmented CAD environment for assembly design. The goals of the project are to: (1) identify representations and issues for the next generation of assembly-related standards; and (2) assist designers with assembly considerations throughout the phases of a product's design - from conception to the final process plan development. OpenADE's open architecture provides standard interfaces that allow it to link to commercial and non-commercial design tools, parametric design systems, virtual reality environments, assembly analysis tools, and assembly process planners. The OpenADE project has explored issues relating to knowledge representation, virtual reality, assembly-level tolerances, constraint-based specifications, and assembly process management. We briefly discuss the representational issues addressed in OpenADE.

Rajan et. al. [5; 6] have primarily emphasized the descriptions of mating constraints between parts (e.g., the description of relative motions between components, fit requirements, and joint strength requirements) in addressing assembly representation issues. In additions to the current ISO and JNC proposed representations, they have explicitly identified and captured the joint kinematics information for assembly modeling during the various stages of design. They propose that some of these kinematics constraints be generated during the conceptual design stage. In addition, other information can generated by propagating/verifying this information during the preliminary and detailed design stages.

During the conceptual design stage, the designer may have little information to fully characterize a joint. Rajan et. al. [5; 6], suggest for rigid joints to describe the type of joint (welded, brazed, soldered, boarded, integral fastening, etc.), and some process parameters to capture pre- and post- process material characteristics (like strength, hardness, ductibility, surface properties, post-heat treatment and machining requirements, acceptable variations, etc.). For kinematic joints, one can specify the desired range of motion and motion profiles, joint location and orientation, and essential surface mating constraints in order to accomplish the desired kinematic joints (revolute, planar, cylindrical, prismatic, screw, spherical, surface, curve, gear, and rack and pinion). As the assembly process (i.e., the product development process) evolves, some more important process constraints are generally specified. The assembly representation model could be captured and used during the planning process. The mating constraints, joint attributes, and a relational structure (specifying positions and orientations of components in the assembly) explicitly embedded within the hierarchical assembly structure provide comprehensive assembly information and are useful for capturing information generated during the detailed design stage.

The Virtual Assembly Design Environment (VADE) [12] acts as an immersive CAD agent for OpenADE. Washington State University developed VADE under a grant from NIST to investigate the use of virtual reality for assembly design. VADE combines advanced CAD/CAM software with the latest in virtual reality technology to produce an environment that allows engineers to virtually assemble a series of components. This produces an augmented CAD system that provides the user with a fully immersive 3D environment for assembly design and analysis.

2.3.2 Design for Tolerancing of Electro-mechanical Assemblies Project

Although Rajan et al stress that the assembly information model should be constructed in such a way that it would be useful in all phases of design (conceptual, preliminary and detailed), their proposed model primarily captures mating constraint information generated during the detailed design stage. As discussed before, important mating constraints, joint attributes, and a relational structure are explicitly embedded within the hierarchical assembly structure. The joint kinematic information and component degrees of freedom are readily available from the assembly model.

The design for tolerancing of electro-mechanical assemblies project group [3; 13] advocates a more general and unified assembly representation scheme for the proactive uses in conceptual and detailed design phases. This scheme includes function, behavior and tolerance information models, along with other assembly information, i.e., 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 Function-Assembly-Behavior (FAB) 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 FAB model and the Design for Tolerancing (DFT) framework use both top-down and bottom-up approaches to product design. The FAB model is defined in such a way that the various definitions and concepts concerning tolerance already in use, as well as currently evolving standards, can be easily incorporated into the model. The FAB model can assist the standards community in specifying the representation for assembly models, tolerance, and assembly level tolerancing. The class structure (Function, Behavior, Artifact) defined in FAB helps in unifying the various current and evolving definitions, concepts, and standards.

2.3.3 NIST Core Product Model

An integrated NIST Core Product Model (NIST-CPM) [4] has been developed to unify and integrate product or assembly information. The NIST-CPM provides a base-level product model that is: not tied to any vendor software; open; non-proprietary; expandable; independent of any one product development process; 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 [8] *class* and *association class*, respectively.

Figure 3 illustrates the principal entities of the NIST-CPM. An **Artifact** refers to a product or one of its components. It is the aggregation of **Function**, **Form** and **Behavior**. **Form** is the aggregation of **Geometry** and **Material**. In addition, an **Artifact** has attributes of **Specification** and **Feature**. **Specification** refers to the general information that contains all the design requirements pertaining to the artifact's function or form. **Feature** represents any information in the **Artifact** that is an aggregation of **Function** and **Form**; purely geometric constructs are not treated as features in the NIST-CPM.. For more information on the NIST-CPM, including the relationships (associations) defined between the classes shown, please refer to [4]. The proposed assembly model in this report is based on the NIST-CPM.

2.3.4 Other Systems

In the following subsections, we discuss several related assembly models. First, we discuss the model proposed by the MOKA [14] (Methodology and tools Oriented to Knowledge-based engineering Applications) consortium. This is followed by a brief review of other work conducted at various academic institutions.

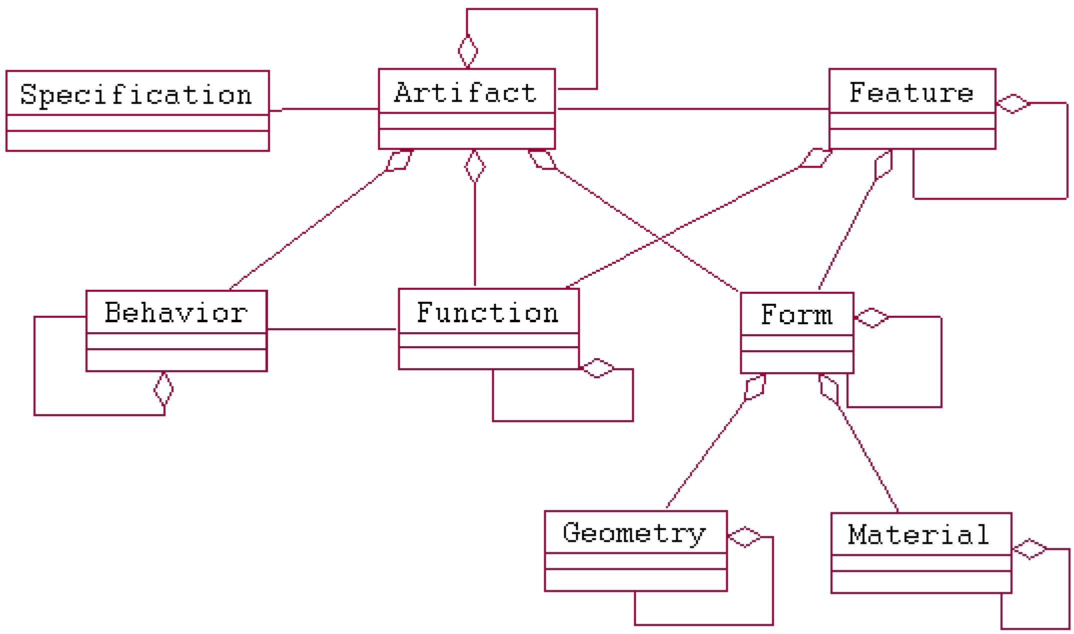


Figure 3: Entities in the Core Product Model (relationships among entities not shown)

MOKA Consortium

The ESPRIT funded project known as MOKA [14] provides two models: (1) an informal model based on a structured, natural language representation of engineering knowledge using some pre-defined forms; and (2) a formal model using UML based graphical, object-oriented representation of engineering knowledge for artifact descriptions. The MOKA methodology intends to synchronize an artifact's lifecycle development process that includes roles, activities, artifact structure, etc., with the whole knowledge-based engineering (KBE) application development lifecycle. Next it covers the entire gamut of engineering knowledge related to an artifact. Knowledge is grouped into two distinct categories: 1) product knowledge which is the knowledge about the physical entity being designed; and 2) process knowledge which is the knowledge about the steps taken to design the product.

MOKA uses a term "Product Model" to describe a model of the entire product family (e.g., Airbus' aircraft product family could be represented as a MOKA Product Model). The MOKA Product Model supports five distinct views of a product: *structure, function, behavior, technology,* and *representation*. These views represent different perspectives of the underlying product model and are as described below.

- *Structure* defines the hierarchical decomposition of a product's structure into parts, assemblies, and features. It can be either a physical, logical or conceptual structure at any stage of the design.

- *Function* defines the functional decomposition of the product, and principles of solutions.

- *Behavior* includes a state model of the various states of a product, and the transition from one state to another.

- *Technology* includes materials and manufacturing process information.

- *Representation* includes any other user defined technological information including some other geometrical information.

In short, the MOKA product representation model is similar in some aspects to the FAB and NIST-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 constraints included in MOKA cannot handle mating, assembly, and parametric constraints. The proposed OAM model can handle these types of constraints, in addition to the constraints described in MOKA. These are essential for assembly, kinematics, and tolerance representations. Representations of the physical structure are supported within MOKA by the Representation View, which includes Geometry and Finite Element Method (FEM). A separate class structure for FEM may be useful in design-analysis integration. However, it is presently being used in MOKA more as a place holder.

There are some academic systems that offer limited facilities for representing assembly information. One such system developed by Whitney and Mantripragada [15] represents the high-level assembly information as "key characteristics". The chains of dimensional relationships and constraints in a product are handled by the so-called Datum Flow Chain concept. Another system called Genesis [16] focuses on representing complex assemblies. The system of van der Net [17] focuses on designing assemblies taking into account requirements from the assembly process planning phase; the goal is 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 relationships on and between the assembled parts.

3 UML Representation of the OAM Assembly

We discuss in detail the class definitions and structure of the proposed OAM assembly model. We use UML notation and diagrams to explain the assembly model.

3.1 Overview

This section introduces the OAM assembly model schema, represented as a UML class diagram. The assembly model's class diagram is made up of multiple sub-diagrams, each of which is represented by a Package. The concept of a UML Package is that of a collection of multiple classes [8] that can be used as a namespace. The following section defines the main schema of the assembly model which describes the relationships between **Assembly, Artifact, AssemblyAssociation, Part,** and **OAMFeature.** In the succeeding sections we describe the class diagrams of **AssemblyAssociation, ArtifactAssociation, OAMFeature, AssemblyFeature, Tolerance, and AssemblyFeatureAssociationRepresentation.** The lower level class hierarchies of these classes are also explained. The instance diagrams are provide in Section 4. These diagrams will further help illustrate the concepts presented here.

3.2 Main Schema of the Assembly model

The main assembly model schema is shown in Figure 4. The model incorporates information about assembly relationship and component composition (*part-of*). The assembly relationship is represented by a class named **AssemblyAssociation**. This class is described in detail in the package **AssemblyAssociation,** shown at the bottom of the diagram and described in Section 3.3

The component composition of the assembly is modeled using the *part-of* relationship. An **Assembly** is decomposed into sub-assemblies and parts. A **Part** is the lowest level component which cannot be further decomposed. The diagram shows that an assembly or a subassembly is made up of at least two parts.

Each assembly component, whether subassembly or part, is made up of one or more features, represented in the model by the **OAMFeature** class, where OAM refers to Open Assembly Model. The details of the **OAMFeature** definition are described in the package **OAMFeature**. The **Assembly** and **Part** classes are sub-classes of the NIST-CPM **Artifact** class, inheriting its definition. **OAMFeature** is a sub-class of the NIST-CPM **Feature** class.

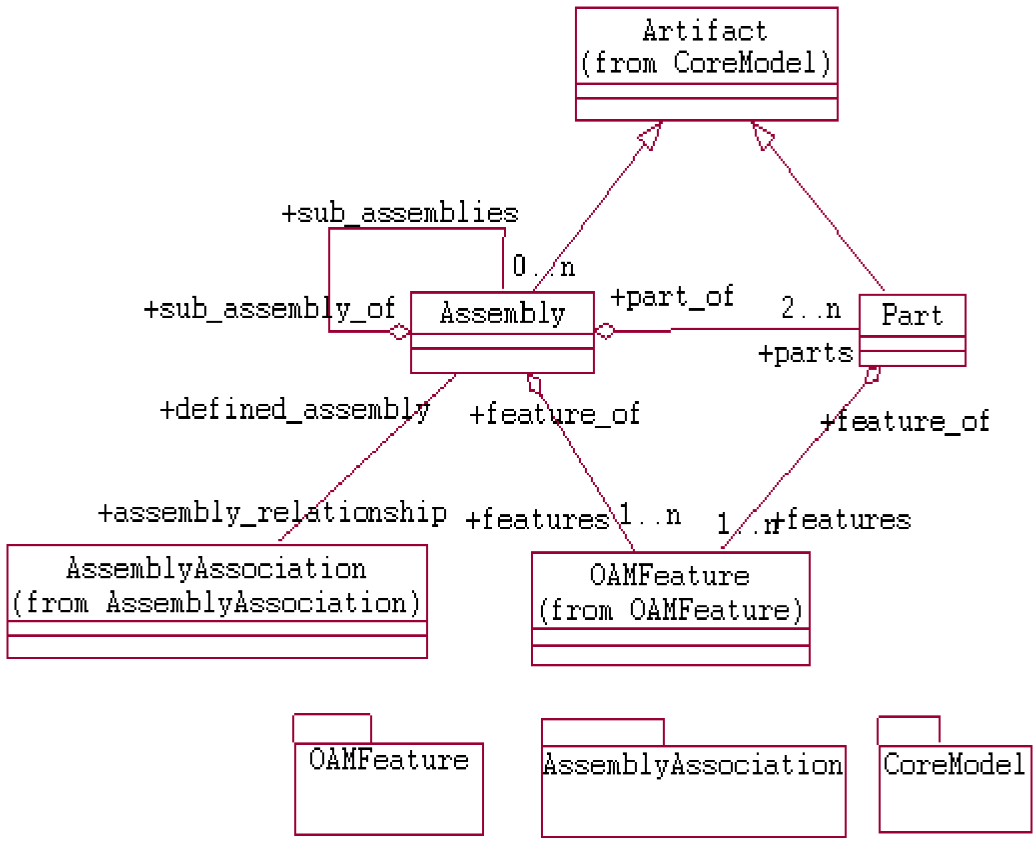


Figure 4: Main Schema of Assembly

3.3 Assembly Association

The package **AssemblyAssociation,** shown in Figure 5, represents the component assembly relationship of an assembly. It is the aggregation of one or more **Artifact Association**s. The class **ArtifactAssociation** is defined in the package **ArtifactAssociation**.

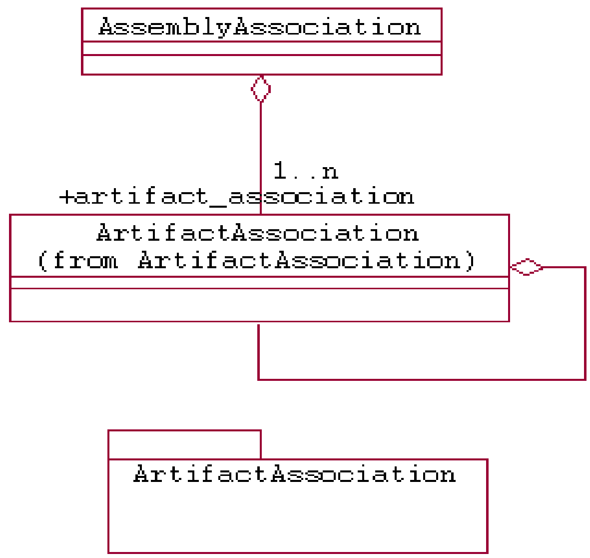


Figure 5: Assembly Association

3.4 Artifact Association

Figure 6 shows the package **ArtifactAssociation** which describes the assembly relationships between artifacts. 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 an artifact is to be fixed in space for anchoring the entire assembly with respect to the ground. It can also occur when kinematic information between an artifact at an input point and the ground is to be captured. Such cases can be regarded as relationships between the ground and an artifact. Hence, we allow the artifact association with one artifact associated in these special cases. In Figure 6, this situation is denoted by the multiplicity *1..** at the **Artifact** end of the association with **ArtifactAssociation**. The detailed information of **ArtifactAssociation** is defined using assembly features, assembly feature associations, and assembly feature association representations, which will be described later in Section 3.6.

The **ArtifactAssociation** is further specialized into the following classes: **PositionOrientation**, **RelativeMotion** and **Connection**.

> **PositionOrientation** represents the relative positions and orientation between two or more artifacts which are not physically connected to each other. This entity is

used to describe constraints on the relative position and orientation of two or more artifacts.

RelativeMotion represents the relative motions between two or more artifacts which are not physically connected to each other directly. This entity is used to describe the constraints on the relative motions between two or more artifacts. For instance, the relative motion of a robot hand against the base of a robot can be represented by this class. In this case, the robot hand is not connected directly to the base.

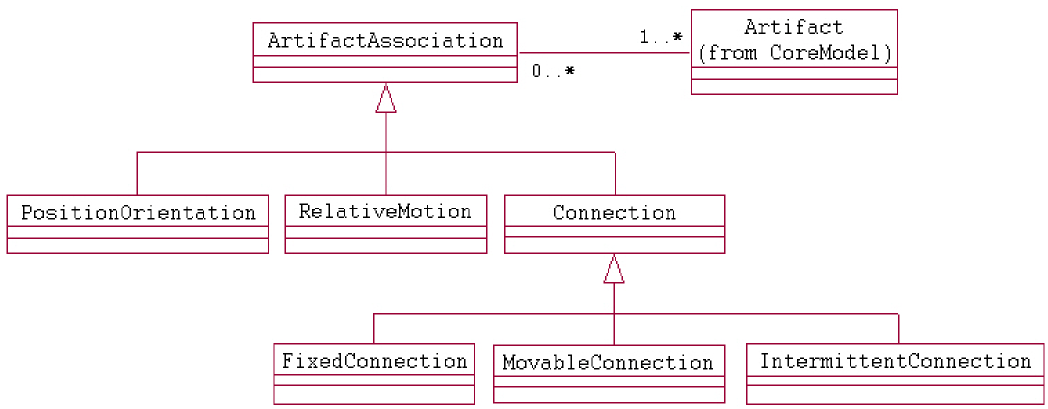


Figure 6: Artifact Association

Connection represents the connection between artifacts which are physically connected with each other. As Figure 6 shows, the **Connection** is further specialized as **FixedConnection**, **MovableConnection** or **IntermittentConnection**.

FixedConnection represents a connection in which the participating artifacts are physically connected and fixed. It is used to describe the type and/or properties of fixed joints. For example, attachment (fastening) with screw, pin, or rivet, bonding such as welding, adhering, brazing, or soldering, and fitting (clearance fit, tight or interference fit) are examples of fixed connections.

MovableConnection represents the connection in which the participating artifacts are physically connected and movable with respect to one another. It is used to describe the type and/or properties of kinematic joints. Typical examples of movable connections are kinematic pairs such as shaft-bearing joints or gear joints.

IntermittentConnection represents the connection in which the participating artifacts are physically connected only intermittently. It is used to describe the type and/or properties of intermittently-connected components. For example, limit switches, contacts using bimetal, or pressure valves with springs are examples of intermittent connection.

3.5 OAMFeature

The class **OAMFeature** is a sub-class of the **Feature** class defined in NIST-CPM. It inherits the function and behavior information from **Feature**. **OAMFeature** has tolerance information which is represented by the class **Tolerance**. **OAMFeature** has two sub-classes: **AssemblyFeature** and **CompositeFeature**.

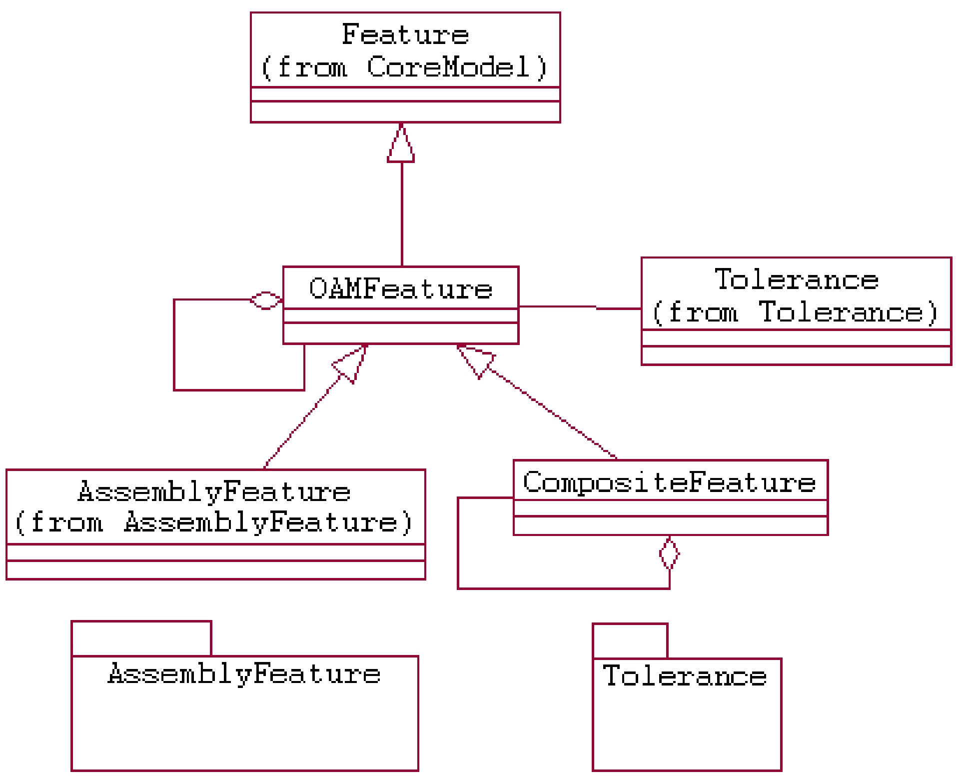


Figure 7: OAMFeature

CompositeFeature represents a composite feature that can be decomposed into multiple simple features.

AssemblyFeature specifies the relationships between a pair of assembled components (e.g., geometrical entities such as faces, edges or vertices of any parts of entities). It is a portion of a constituent and is used for defining the connectivity relationship between constituents of an assembly model.

3.6 Assembly Feature

Figure 8 shows the relationship between classes that are related to assembly features. The class **AssemblyFeatureAssociation** refers to the assembly relationship between one or more assembly features. This assembly relationship is represented by the class **AssemblyFeatureAssociationRepresentation**. The diagram also shows that the **ArtifactAssociation** is the aggregation of **AssemblyFeatureAssociation**. Detailed description of these classes are given below.

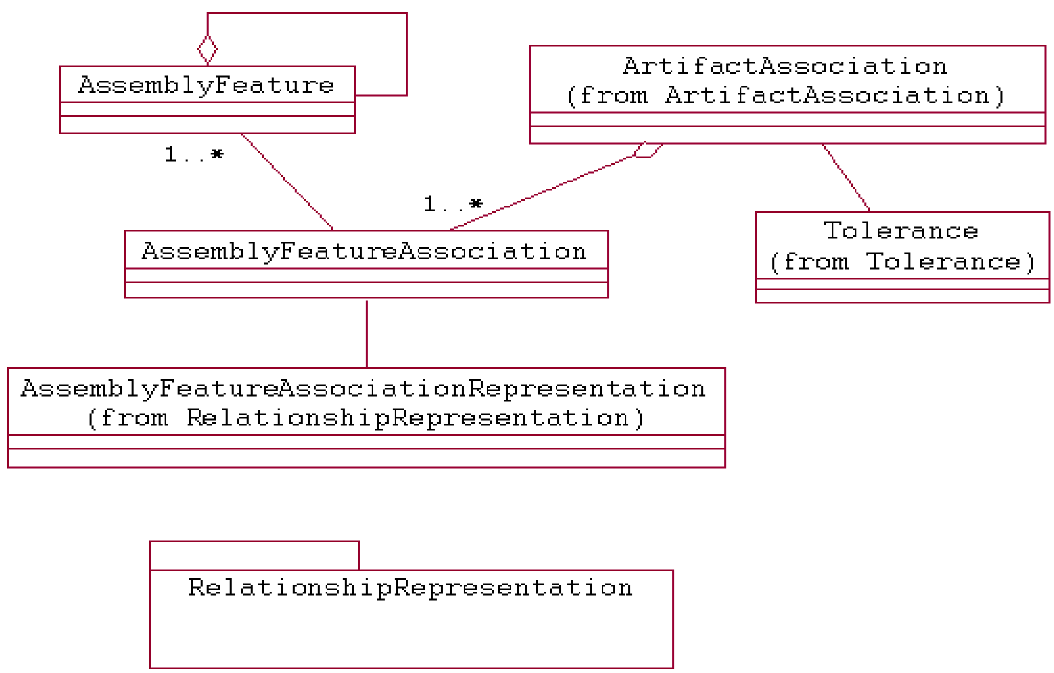


Figure 8: Assembly Feature

AssemblyFeature, a sub-class of **OAMFeature,** is defined to represent 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. A 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, screws and nuts, spheres, cones, and toruses as assembly features.

AssemblyFeatureAssociation represents the association between mating assembly features through which relevant artifacts are associated. 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 as shown in Figure 8. This relationship is also identified by the multiplicity symbol *1..** (see Figure 8). Any assembly feature association relates in general to two or more assembly features. However, as in the special case mentioned earlier (see Section 3.4) where an artifact association involves only one artifact, it may involve only one assembly feature when the relevant artifact association has only one artifact. Hence, the **AssemblyFeature** end of the association with **AssemblyFeatureAssociation** in Figure 8 is denoted by the multiplicity symbol *1..*,* not by the multiplicity symbol *2..*.*

3.7 Assembly Feature Association Representation

The specific contents of any assembly feature association is described by the **AssemblyFeatureAssociationRepresentation** (see Figure 9). The assembly feature association representation is an aggregation of parametric assembly constraints, a kinematic pair, and/or a relative motion between assembly features as shown in Figure 9.

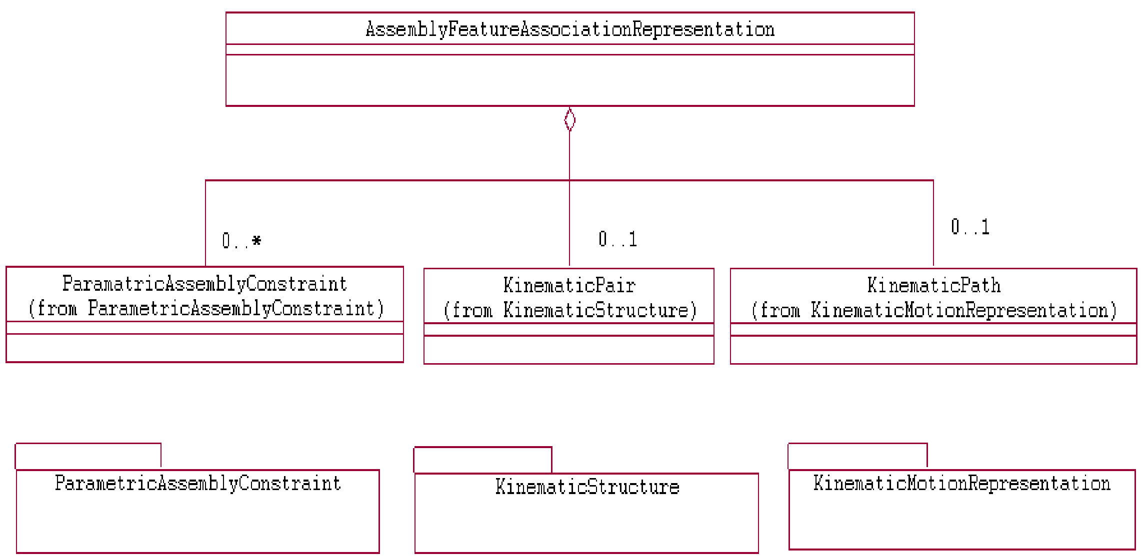


Figure 9: Assembly Feature Association Representation

ParametricAssemblyConstraint represents the parametric assembly constraints of assembly feature associations. It is used to specify explicit geometric constraints between artifacts of an assembled product. The parametric assembly constraints are intended to control the position and orientation of artifacts in an assembly. Parametric assembly constraints are defined in ISO 10303 Part 108 (under ballot) and a brief description is given in the next section.

KinematicPair defines the kinematic constraints between two adjacent artifacts (links) at a joint. The kinematic structure schema in ISO 10303 Part 105 defines the kinematic structure of a mechanical product in terms of links, pairs, and joints. The kinematic pair represents the geometric aspects of the kinematic constraints of motion between two assembled components. The description of the kinematic structure is given later in Section 3.9.

KinematicPath represents the relative motion between artifacts. The kinematic motion schema in ISO 10303 Part 105 defines kinematic motion. It is also used to represent the relative motion between artifacts. Details of this are provided in Section 3.10.

In Section 3.8, detailed descriptions of the parametric assembly constraint, kinematic pair, and kinematic path are provided. These three packages defined in the ISO standards are of considerable complexity, and thus the descriptions below are not exhaustive. Only the necessary portions are modeled using UML to explain how they can be utilized to represent electro-mechanical assemblies in conjunction with the proposed UML model. Note that we have introduced some new classes and used different class names from the original ones defined in ISO standards for increased clarity.

3.8 Parametric Assembly Constraints

The classes in Figure 10 represent parametric assembly constraints. The parent class of **Parametricassemblyconstraint** defines the assembly constraints (position and orientation) between two parts in an assembly. From this class we specialize specific types of assembly constraints, as shown in Figure 10. The constrained geometric elements can be a point, a line, a circle, a plane, a cylindrical surface, a conical surface or a spherical surface of the constrained parts. Each type of assembly constraints applies to specific types of geometric elements. One of the two elements may be the reference element when applying the constraint.

Parallel specifies that two geometric elements are mutually parallel. **ParallelWithDimension** constrains two parallel elements with a distance. **SurfaceDistanceWithDimension** specifies the distance between two surfaces.

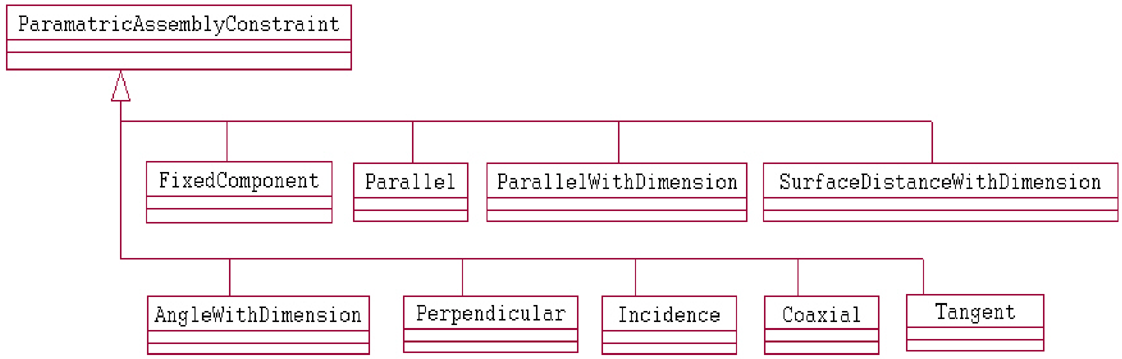


Figure 10: Parametric Assembly Constraints

The angle between two elements is constrained by **AngleWithDimension**. **Perpendicular** constrains two elements to be mutually perpendicular. **Incidence** represents that one of two elements is included in the other when it is considered as a point set. **Coaxial** means that the axes of the two elements are coincident. **Tangent** specifies that two elements are tangent to each other. Finally, **FixedComponent** is provided to fix the position and orientation of one part thus anchoring its own assembly in space. This constraint can be viewed as an exceptional case as it involves only one part; implicitly, the other part is the ground (i.e., the world coordinate system).

3.9 Kinematic Pair

KinematicPair defines the kinematic constraints between two adjacent artifacts (links). Figure 11 shows the relationship between classes that are used to represent the kinematic constraints of kinematic pairs. Specific types of kinematic pairs are specialized from **KinematicPair**, **PairValue**, and **PairRange**.

KinematicPair defines the kinematic constraints between two adjacent links (artifacts) at a joint. **PairRange** specifies the allowable configuration range of the two links in the form of upper and lower bounds. **PairValue** specifies the current configuration (value) of the two links between the two bounds. **PairFrame** represents a coordinate system attached to a link. A kinematic pair needs two coordinate systems to describe its kinematic behavior. The two coordinate systems are attached to the two relevant links, respectively. Thus, the multiplicity at the **PairFrame** end is shown as 2 in Figure 11.

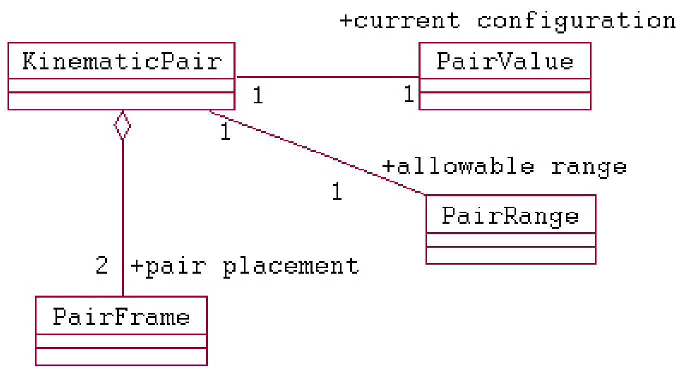


Figure 11: Kinematic Pair

As mentioned above, specific classes of kinematic pairs are derived from the classes shown in Figure 11. Figure 12 shows the derived types of kinematic pairs. The kinematic pairs cover the following types: revolute, prismatic, screw, cylindrical, spherical, universal, planar, gear, rack and pinion, unconstrained, fully constrained, point on surface, sliding surface, rolling surface, point on planar curve, sliding curve, rolling curve.

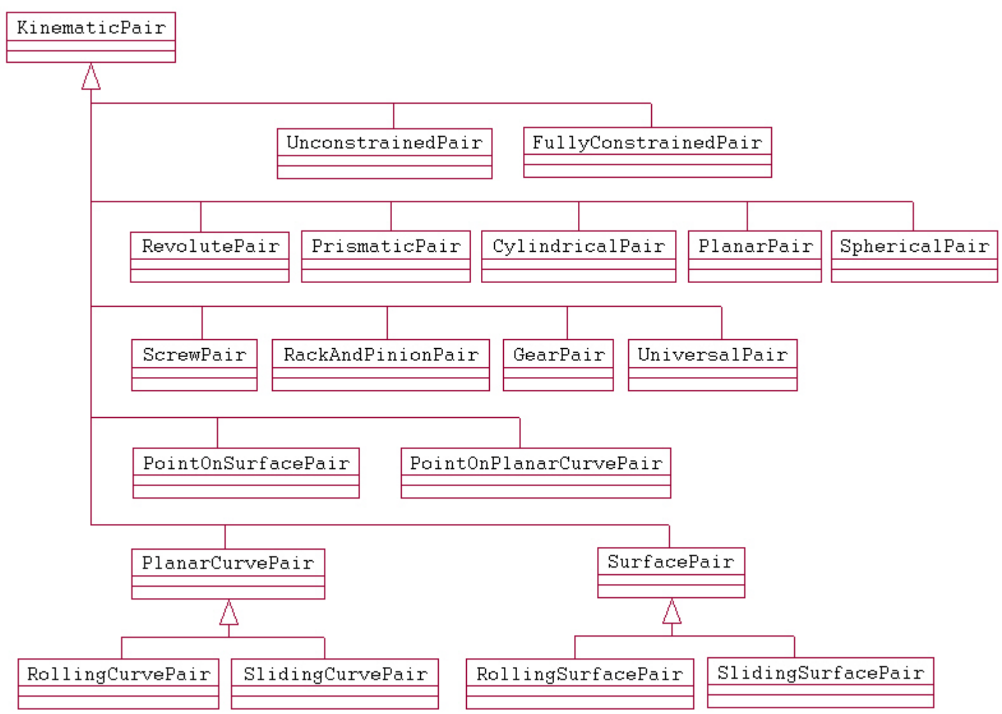


Figure 12: Derived Kinematic Pairs

Each kinematic pair has two specific classes derived from **PairValue** and **PairRange** respectively, to represent the current configuration (value) and the allowable range of the pair variable. For example, the revolute pair class diagram in Figure 13 shows the derived classes **RevolutePair**, **RevolutePairValue**, **RevolutePairRange**, and their associations.

The classes **RevolutePairValue** and **RevolutePairRange** are specialized, respectively, from **PairValue** and **PairRange**. Other types of kinematic pairs also have specific type of derived classes defined for their own purpose.

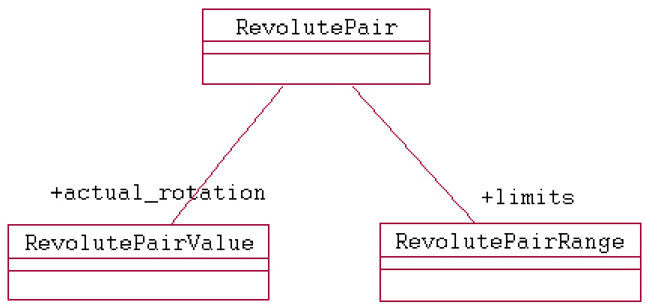


Figure 13: Revolute Pair Class Diagram

3.10 Kinematic Path

KinematicPath, shown in Figure 14, provides the description of kinematic motion. It is the aggregation of path elements along which the motion is to take place. A path element can specify different types of paths, as explained below. Note that since the kinematic path is composed of a set of path elements, it can describe a composite path as well as a simple path.

PathElement is a path segment with two path nodes, which represent the "from" node and "to" node, respectively. **PathNode** is used to define the start and end locations of a path. At each path node, the position and rotation of a frame along the "path" need to be defined. This linear transformation is defined by **Transformation,** which can is expressed by a 4×4 homogeneous transformation matrix.

The path element is further classified into different types of motion according to the intermediate motion characteristics between the "from" and "to" nodes, as shown in Figure 14. **PointToPointPath** allows free motion between the two nodes; it does not have a prescribed interpolation and leaves the determination of the intermediate motion to specific applications. **LinearPath** incorporates piecewise linear interpolation between the beginning and end nodes of the path element. **CircularPath** describes a circular arc path which passes through the beginning and end points of the path element, and also a third point (via point) supplied separately. **CurveBasedPath** describes the path using a predefined trajectory curve.

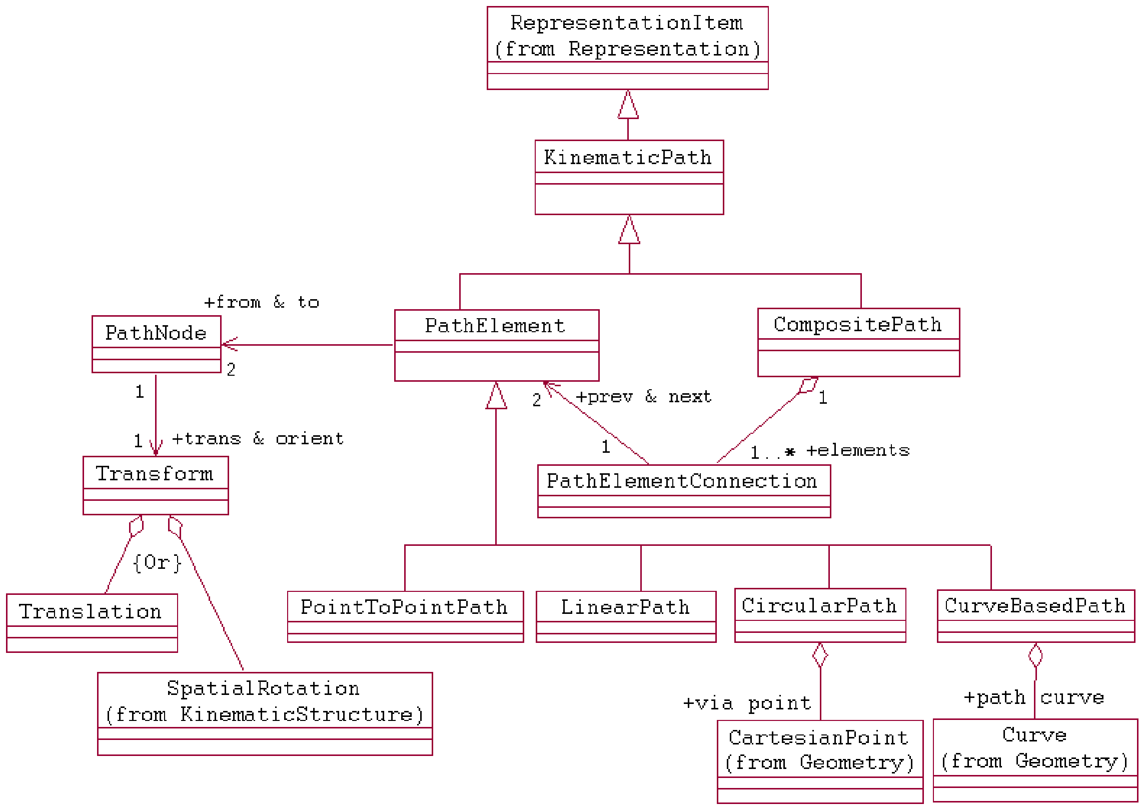


Figure 14: Kinematic Path

3.11 Tolerance

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 subassembly 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. During the design of an assembly, both the assembly structure and the associated tolerance information evolve continuously; significant gains can thus be achieved 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 conceptual product 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 describe this in Figure 15 .

DimensionalTolerance typically controls the variability of linear dimensions that describe location, size and angle, also known as tolerancing of perfect form. This class is included in the model to accommodate the ISO tolerance standard.

GeometricTolerance is the general term applied to the category of tolerances used to control shape, position, and runout. It enables tolerances to be placed on attributes of features, where a feature is one or more 'pieces of part surface'; feature attributes include size (for certain features), position (certain features), form (flatness, cylindricity, etc.), and relationship (e.g. perpendicular-to).

FormTolerance is applicable to single features or elements of single features.

> **ProfileToleranc**e is applicable to profiles. A profile is the outline of an object in a given plane. Profiles are formed by projecting a three-dimensional figure onto a plane or by taking a cross section through the figure.
>
> **RunoutTolerance** is a composite tolerance used to control the functional relationship of one or more features of a part to a datum axis.

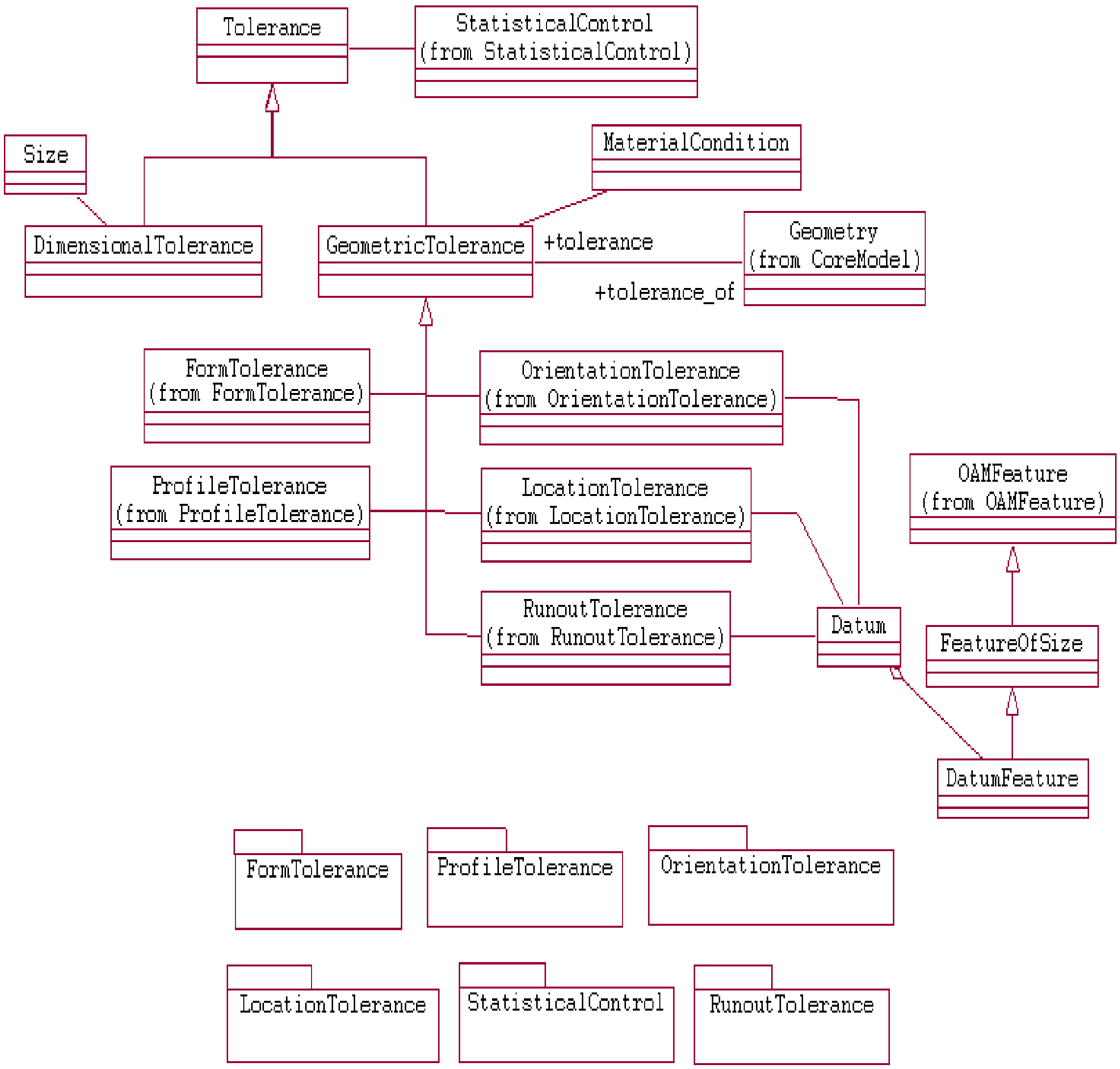


Figure 15: Tolerance Package

OrientationTolerance is applicable to feature orientations. Angularity, parallelism, perpendicularity, and in some cases, profile are orientation tolerances applicable to two or more features. These tolerances control the orientation of a feature with respect to another feature.

LocationTolerance deals with the position, concentricity, and symmetry used to control the following relationships:

- center distance between such feature as holes, slots, and tabs;
- location of features as a group, from datum features, such as a plane and a cylindrical surface;
- coaxiality of features; and
- concentricity or symmetry of features.

Datum

Datum is a theoretically exact piece of geometry, such as a point, a line, and a plane, from which a tolerance is referenced.

DatumFeature

DatumFeature is a physical feature that is applied to establish a datum.

FeatureOfSize

A feature that is associated with a size dimension, such as the diameter of a spherical or cylindrical surface and the distance between two parallel planes.

3.11.1 Form Tolerance

Figure 16 presents the form tolerance of a component.

StraightnessTolerance

Straightness is a condition where an element of a surface, or an axis, is straight line. StraightnessTolerance signifies the allowable variations from this condition.

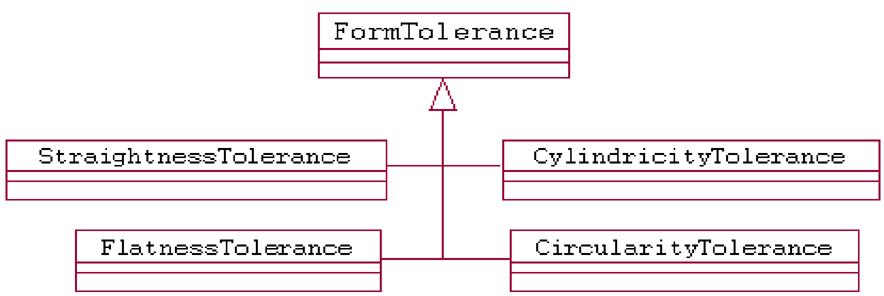


Figure 16: Form Tolerance Package

FlatnessTolerance

Flatness is the condition of a surface having all elements in one plane. **FlatnessTolerance** signifies the allowable variations from this condition.

CircularityTolerance

Circularity is a condition of a surface where, for a:

- feature other than a sphere, all points of the surface intersected by any plane perpendicular to an axis are equidistant from that axis;

- sphere, all points of the surface intersected by any plane passing through a common center are equidistant from that center.

CircularityTolerance signifies the allowable variations from this circularity condition.

CylindricityTolerance
Cylindricity is a condition of a surface of revolution in which all points of the surface are equidistant from a common axis. **CylindricityTolerance** signifies the allowable variations from this condition.

3.11.2 Profile Tolerance

Figure 17 presents the profile tolerance of a component .

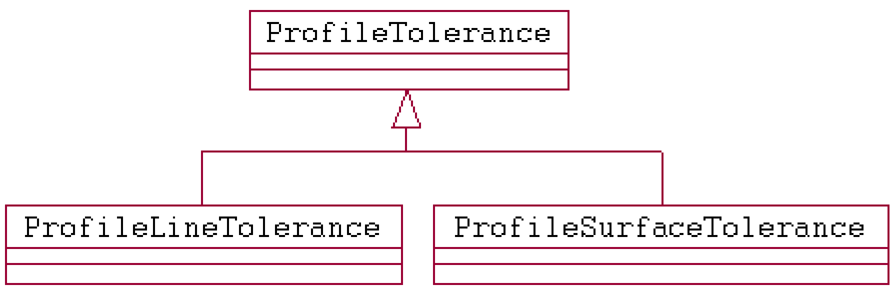


Figure 17: ProfileTolerance Package

ProfileSurfaceTolerance
The tolerance zone established by the profile of a surface tolerance is three-dimensional, extending along the length and width of the considered feature or features.

ProfileofaLine
The tolerance zone established by the profile of a line tolerance is two-dimensional, extending along the length of the considered feature.

3.11.3 Location Tolerance

Figure 18 presents the location tolerance of a component.

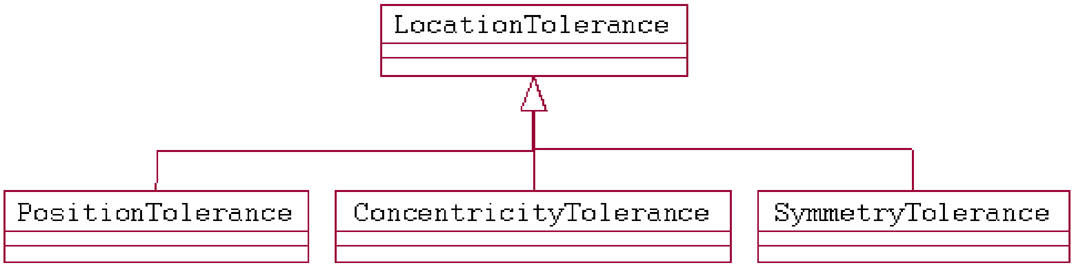


Figure 18: LocationTolerance Package

PositionTolerance defines either :

- a zone within which the center, axis, or center plane of a feature of size is permitted to vary from a true position, or
- a boundary, defined as the virtual condition, located at the true position, that may not be violated by the surface or surface of the considered feature.

ConcentricityTolerance is a cylindrical tolerance zone whose axis coincide with the axis of the datum feature**.**

SymmetryTolerance
Symmetry is that condition where the median point of all opposed or correspondingly located elements of two or more feature surfaces are congruent with the axis or center plane of a datum feature.

3.11.4 Orientation Tolerance

Figure 19 presents the orientation tolerances between components.

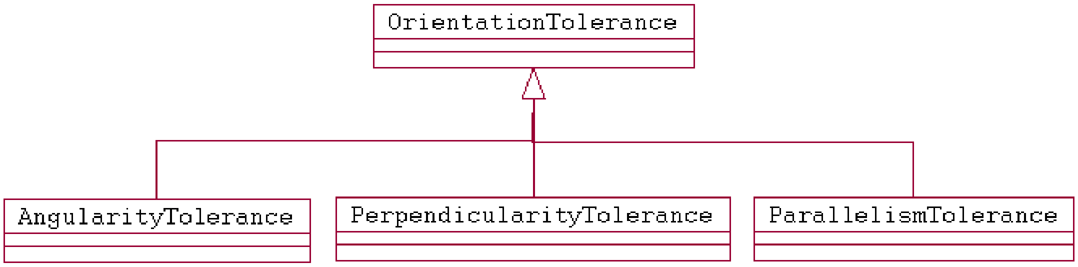


Figure 19: OrientationTolerance package

AngularityTolerance
Angularity is the condition of a surface, a center plane, or an axis at a specified angle from a datum plane or an axis.

ParallelismTolerance
Parallelism is the condition of a surface or a center plane, equidistant at all points from a datum plane; or an axis, equidistant along its length from one or more datum planes or a datum axis.

PerpendicularityTolerance
Perpendicularity is the condition of a surface, center plane, or axis at a right angle to a datum plane or axis.

3.11.5 Runout Tolerance

Figure 20 presents the runout tolerance of a component.

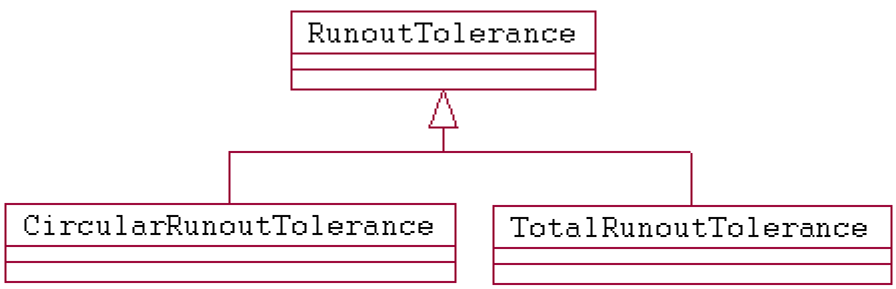


Figure 20: RunoutTolerance Package

CircularrunoutTolerance provides control of circular elements of a surface. The tolerance is applied independently at each circular measuring position, as the part is rotated 360 degrees.

TotalrunoutTolerance provides composite control of all surface elements. The tolerance is applied simultaneously to all circular and profile measuring position, as the part is rotated 360 degrees.

3.11.6 Statistical Tolerance

StatisticalControl
StatisticalControl is a tolerance that requires statistical process controls on the toleranced feature in manufacturing. (See Figure 21).

RegularControl
RegularControl is a statistical tolerance that is defined by the mean and the standard deviation.

DistributedControl
DistributedControl is a statistical tolerance that is defined by a range of distributions of dimensional variations. For example, assume that the tolerance of a 100 mm shaft is ±1 mm. At least fifty percent of the produced shafts should be within the tolerance of ±0.5 mm. The remaining must be within the tolerance of ±1 mm.

Range
Range is the definition of the range of dimensional variations. For example, fifty percent of the produced shafts should be within the tolerance of ±0.5 mm.

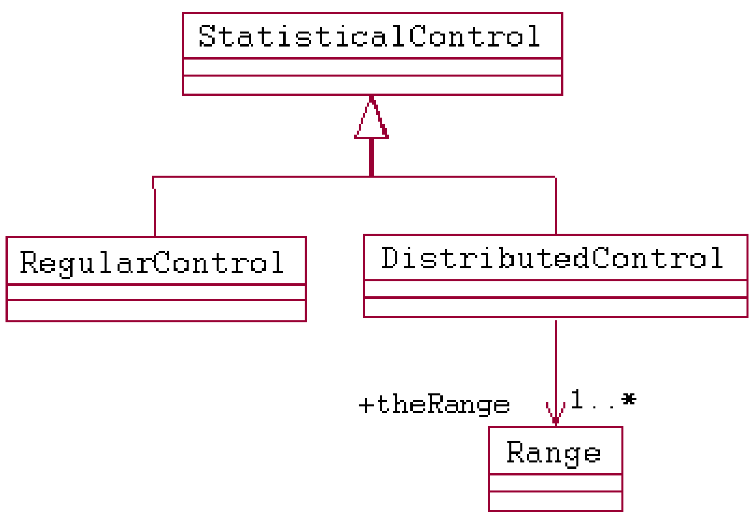


Figure 21: StatisticalTolerance Package

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. The first Section 4.1 lists all the parts in the assembly. Section 4.2 describes the principal hierarchy of the assembly. Section 4.3 explains the assembly relationships such as artifact associations, assembly feature associations, assembly constraints, and kinematic pairs. Section 4.4 describes the kinematic relationships in detail. Finally, Section 4.5 illustrates the tolerances that are needed for assembly tolerance analysis.

4.1 Components in Planetary Gear System

A planetary gear system is used to illustrate our model. The solid model of the planetary gear system is shown in Figure 22.

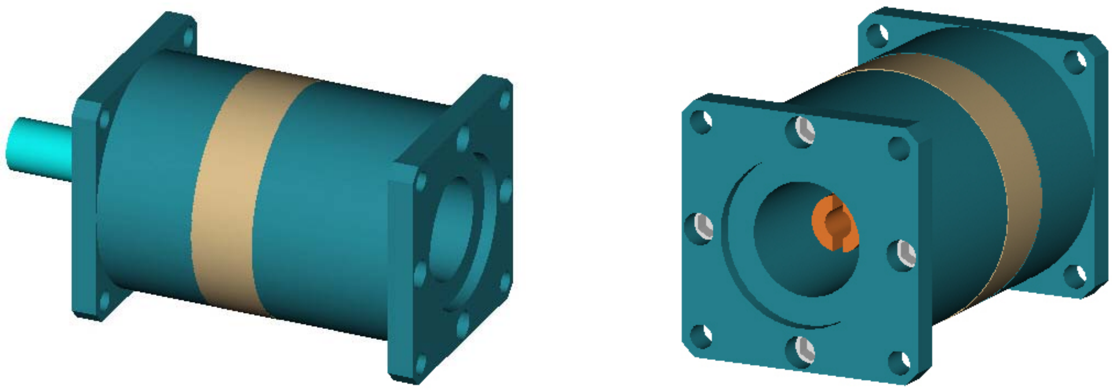


Figure 22: Solid Model of a Planetary Gear

The planetary gear system consists of many components. Figure 23 shows the exploded view of the above solid model.

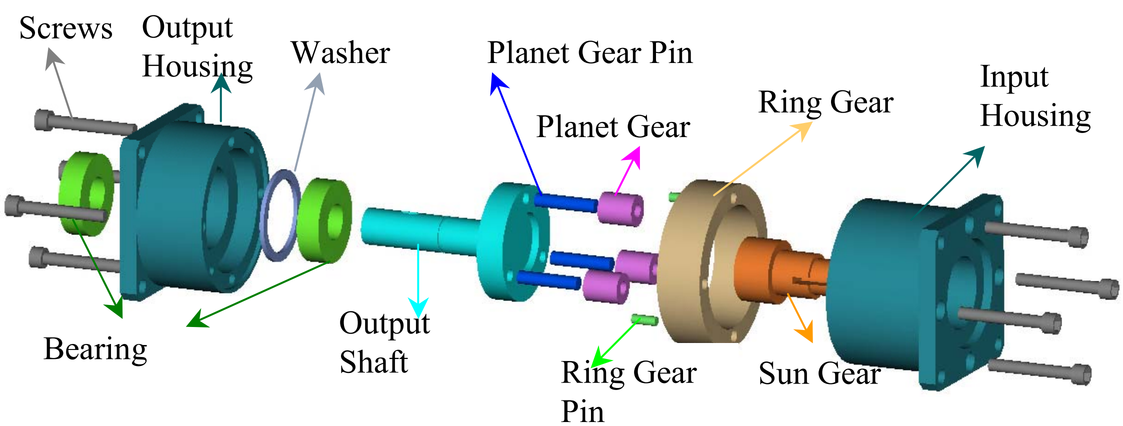


Figure 23: Exploded View of the Planetary Gear Model

The list of all the components of the planetary gear system shown in Figure 23 is given in Table 1.

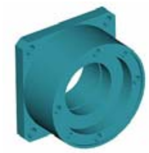
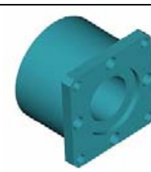
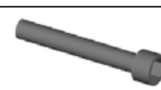
ID	Name	Qty	Functional Description	Graphical Representation
1	Output Housing	1	Covers output shaft and protects the gears and shafts	
2	Bearing	2	Supports the output housing and serves as an interface between the output shaft and the housing	
3	Washer	1	Separate the two bearings inside the output housing	
4	Output shaft	1	Transmits power to the driven device. Also, connects to planetary gears.	
5	Planet gear pin	3	Holds a planetary gear and attaches it to the output shaft	
6	Planet gear	3	Delivers power from the sungear to the output shaft	
7	Ring gear	1	Controls the speed reduction ratio. The planetary gears rotate around it.	
8	Ring gear pin	2	Attaches the ring gear to the output housing	
9	Sungear	1	Transmits the power from input shaft to planetary gears. Input shaft and sungear considered as one part.	
10	Input housing	1	Covers the input shaft and provides protection from environmental contamination.	
11	Screw	8	Fastens the input housing, the ring gear, and the output housing into one assembly.	

Table 1: Part List of the Planetary Gear System

4.2 Assembly Hierarchy

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 two parts, namely, the input-housing and the sungear, and three sub-assemblies, as shown in Figure 24. The three 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; and (3) the planet gear holder assembly that consists of the three planet gears and the planet carrier assembly, which further decomposes into the output shaft and the three planet-gear pins.

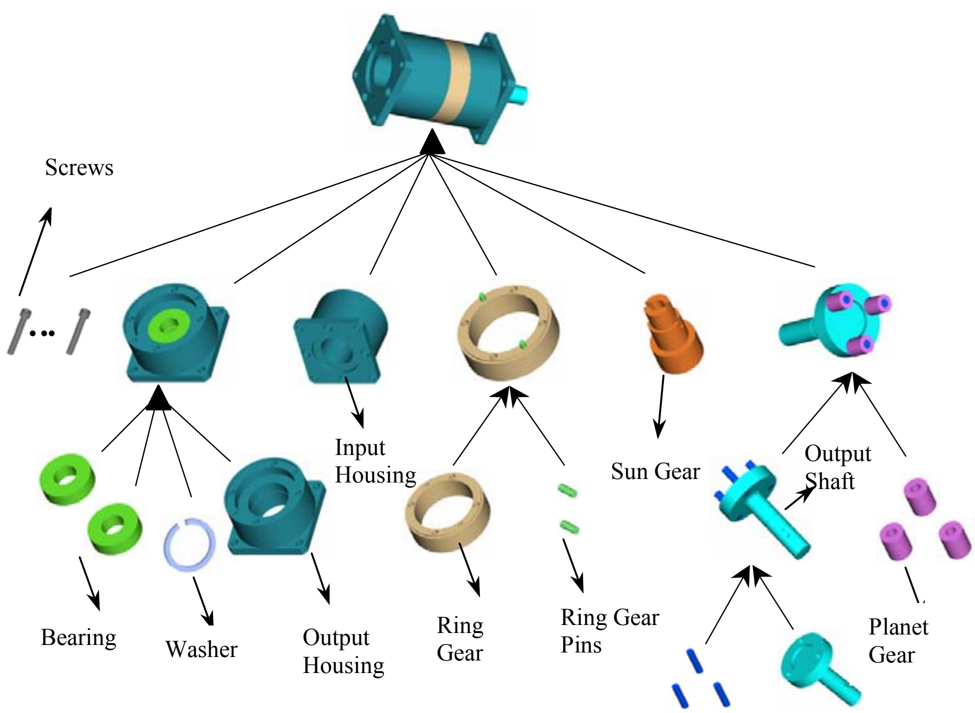


Figure 24: Assembly Hierarchy of Planetary Gear System

. Notice that the hierarchy in Figure 24 is introduced to verify and demonstrate the proposed UML assembly model. The sequence of part assembly descriptions does not imply the actual assembly sequence. The assembly sequencing task is outside the scope of this report.

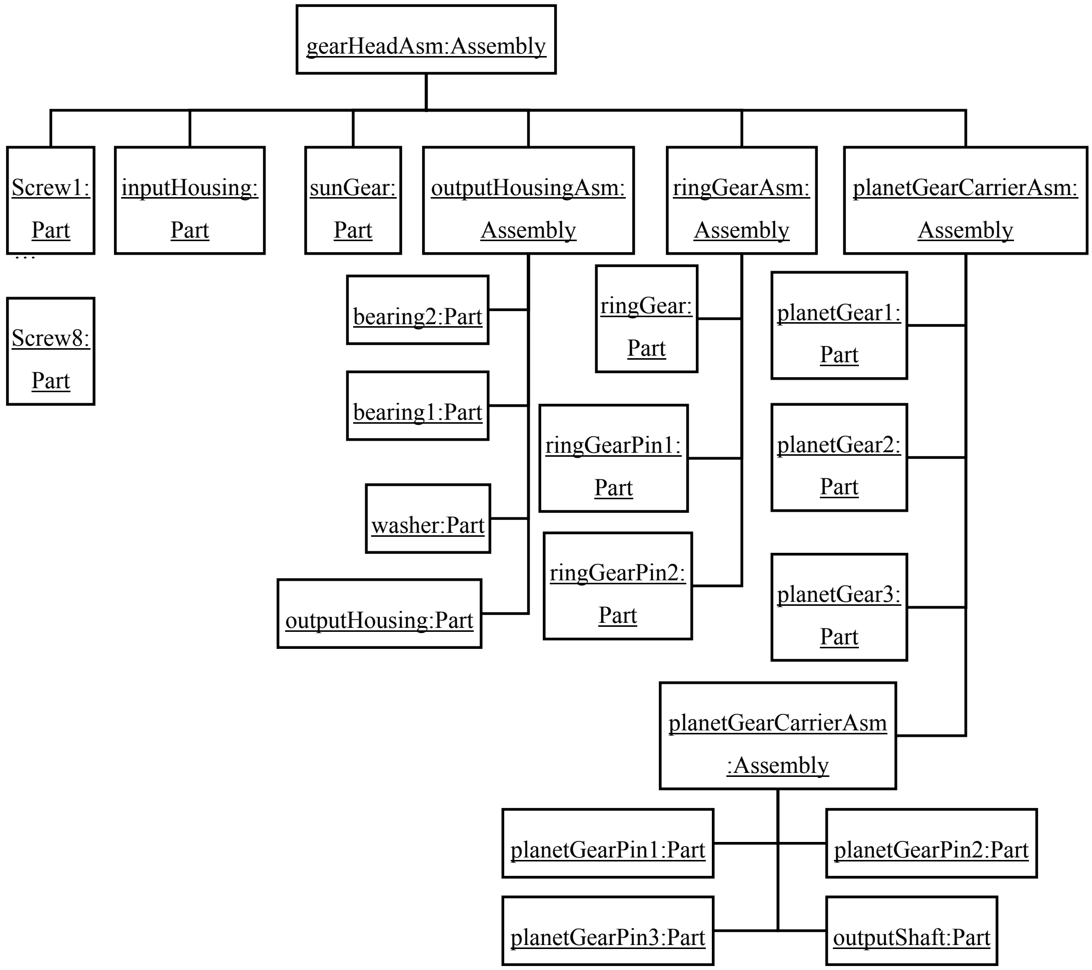


Figure 25: Instance Diagram of Main Assembly Hierarchy

The hierarchical relationships between the components of the planetary gear system can be represented as an instance diagram as shown in Figure 25. The elements are underlined to show that they are instances. The names take the form of "*instance name:class name*". The root node is the entire assembly, the interior nodes are sub-assemblies, and the leaf nodes are component parts.

4.3 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. For the gear head assembly shown in Figure 25, a graph of artifact associations is shown in Figure 26. The dotted lines indicate the artifact associations and the solid lines portray the

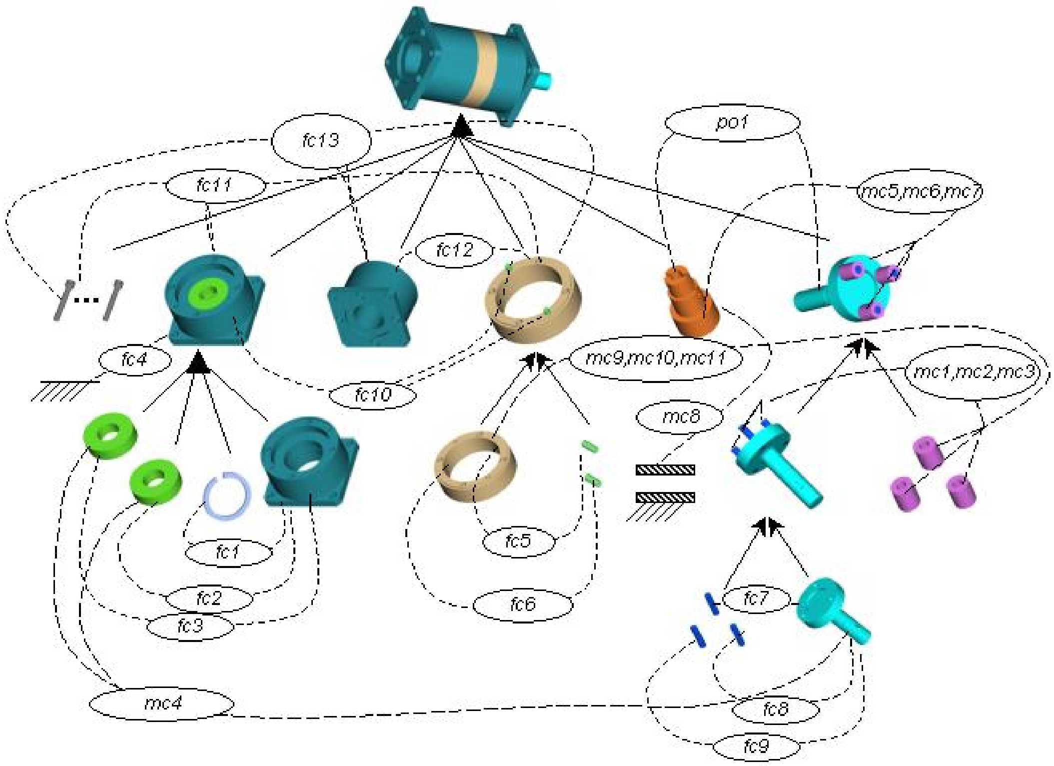


Figure 26: Artifact Associations

hierarchical assembly relationships. The details of the artifact associations shown in Figure 26 are discussed in the following sections. First we explain the assembly relationship for each subassembly, and then we describe the entire assembly.

4.3.1 Output Housing Assembly

Figure 27 describes the subassembly and the output housing (the output end of the planetary gear system). It consists of four parts: bearing 1, bearing 2, washer, and output housing. The washer goes to the inside groove of the output housing. Both bearings (ball bearings) go into the output housing on both sides of the washer by a tight fit. Bearing 1 stays outside, and Bearing 2 stays inside of the planetary gear system.

The assembly relationships between artifacts are shown in Table 2. The first column shows the names of the artifact associations. The second column lists the participating artifacts in the associations. The relevant assembly features, parametric assembly constraints, and kinematic relationships are also listed. The instance diagram of the assembly relationships in Table 2 is depicted in Figure 28. Some class names are denoted with acronyms instead of their long class names (Figure 28).

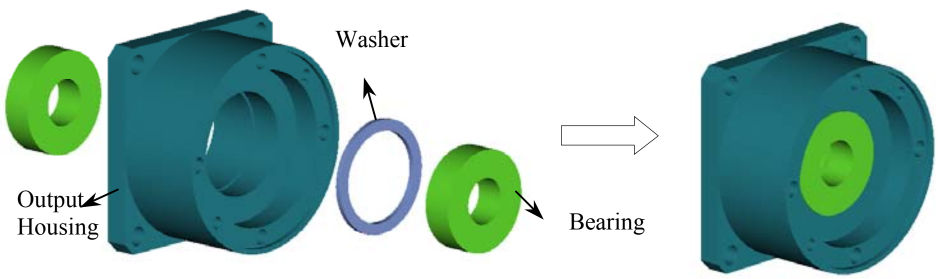


Figure 27: Output Housing Assembly

The artifact associations are instantiated from the class **FixedConnection,** one of the sub-classes of **ArtifactAssociation,** since the participating artifacts in the present subassembly have no relative motion with respect to one another. The associations are named *fc1*, *fc2*, and so forth. The details of the artifact associations are represented by accompanying instances of assembly features (AFs) and assembly feature associations (AFAs) as shown in Figure 28. The assembly feature association representations (AFARs) contain the parametric assembly constraints to position artifacts with respect to a reference (fixed) artifact. We assume here that the output housing has a fixed position and orientation to anchor the entire assembly. Thus, the output housing itself should have a parametric assembly constraint of **FixedComponent.** To hold this information, an instance of **ArtifactAssociation** (fixed connection) *fc4*, with information regarding assembly feature and assembly feature association is provided, as shown in Table 2. Notice that the artifact association *fc4* has only one artifact (the output housing) associated with it, and so does its assembly feature association. The artifact association *fc3* between bearing 2 and output housing will have a similar assembly relationships structure to that of *fc2* and is not depicted in Figure 28 for brevity.

Artifact associations	**Artifacts**	**Assembly features**	**Assembly constraints**	**Kinematic relationships**
fc1	Output housing	All three surfaces of inner groove (groove:AF)	Coaxial Parallel Angle with dimension	No relative motion
	Washer	Surface of outer rim and partial surfaces on both sides (outer Rim:AF)		
fc2	Output housing	Cylindrical surface for the outside bearing seat (bearingSeat1:AF)	Coaxial Parallel Angle with dimension	No relative motion
	Bearing 1	Rim of the bearing (outerRace1:AF)		

Artifact associations	Artifacts	Assembly features	Assembly constraints	Kinematic relationships
fc3	Output housing	Cylindrical surface for the inside bearing seat (bearingSeat2:AF)	Coaxial Parallel Angle with dimension	No relative motion
	Bearing 2	Rim of the bearing (outerRace2:AF)		
fc4	Output housing	Entire subassembly	Fixed	No relative motion

Table 2: Assembly relationships of the output end of the planetary gear system

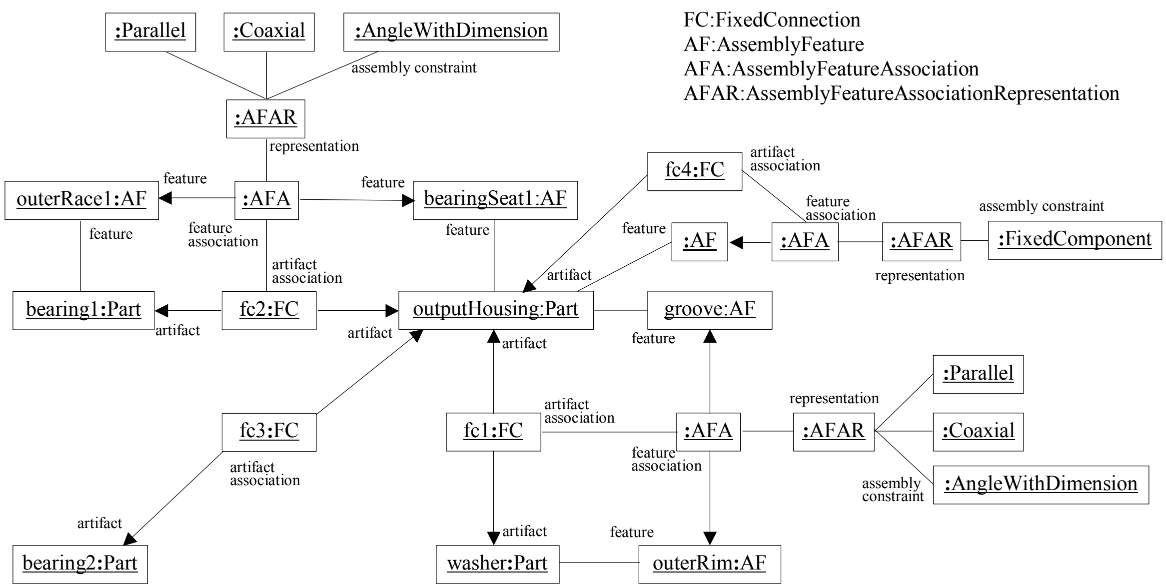


Figure 28: Instance Diagram of Output Housing Assembly

4.3.2 Ring Gear Assembly

The second subassembly is the ring gear assembly, shown in Figure 29. 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.

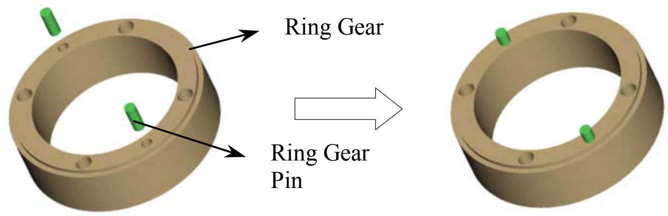


Figure 29: Ring Gear Assembly

The assembly relationships are listed in Table 3, and Figure 30 shows the instance diagram of the current assembly. 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.

Artifact associations	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 3: Assembly Relationships of Ring Gear Assembly

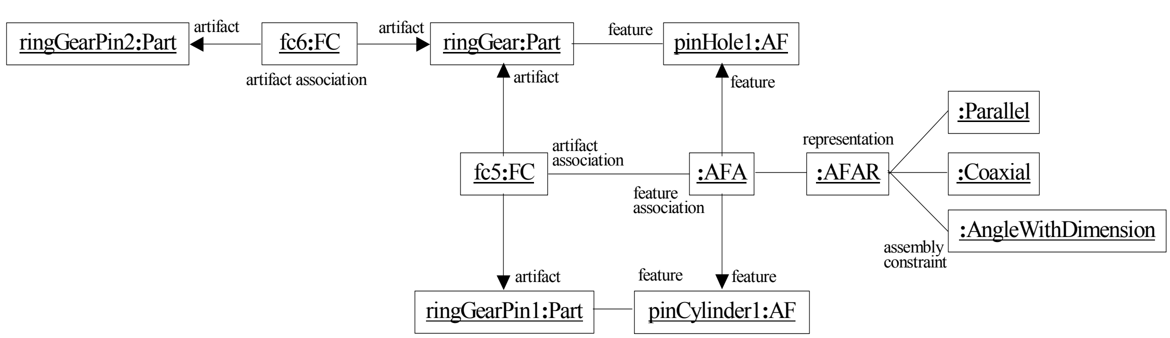


Figure 30: Instance Diagram of Output Housing Assembly

4.3.3 Planet Carrier Assembly

The planet carrier assembly in Figure 31 is comprised of four parts: three planet-gear pins and an output shaft. The three planet-gear pins are assembled with output shaft by a tight fit.

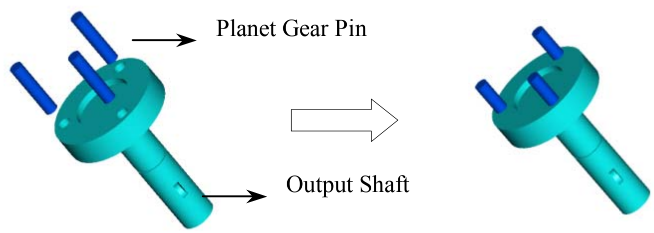


Figure 31: Planet Carrier Assembly

The assembly relationships are listed in Table 4, and the instance diagram is depicted in Figure 32. 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 *fc6*.

Artifact associations	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 4: Assembly Relationships of Planet Carrier Assembly

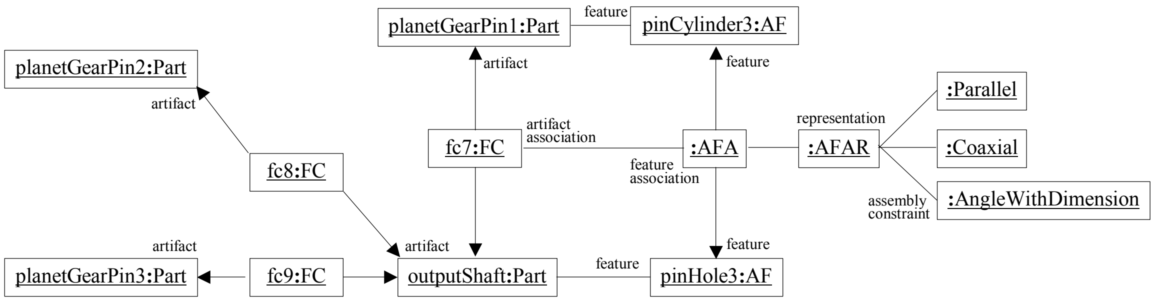


Figure 32: Instance Diagram of Planet Carrier Assembly

4.3.4 Planet Gear-carrier Assembly

The planet gear-carrier assembly shown in Figure 33 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.

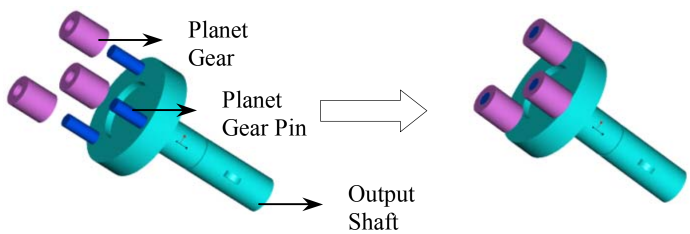


Figure 33: Planet Gear Carrier Assembly

Table 5 lists the assembly relationships. These assembly relationships are illustrated in Figure 34. 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 associations	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 5: Assembly Relationships of Planet Gear-Carrier Assembly

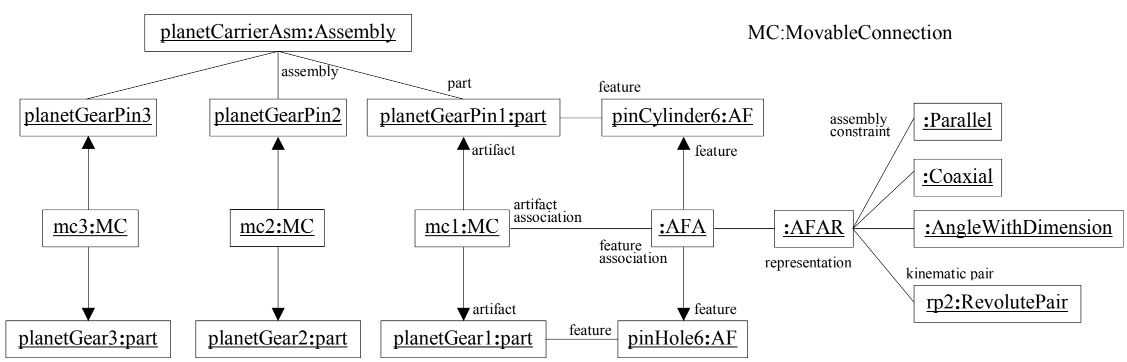


Figure 34: Instance Diagram of Planet Gear-Carrier Assembly

4.3.5 Planetary gear system Assembly

The planetary gear system assembly in Figure 35 consists of five artifacts: two parts and three assemblies. The planet gear-carrier assembly is pushed into the bearings of the output housing assembly. The ring gear assembly is attached to the output housing assembly, being meshed with the three planet gears of the planet gear-carrier assembly. The sungear is also meshed with the three planet gears. The input housing is attached to the ring gear. The details of the assembly relationships are explained for some of the artifacts to avoid repetition.

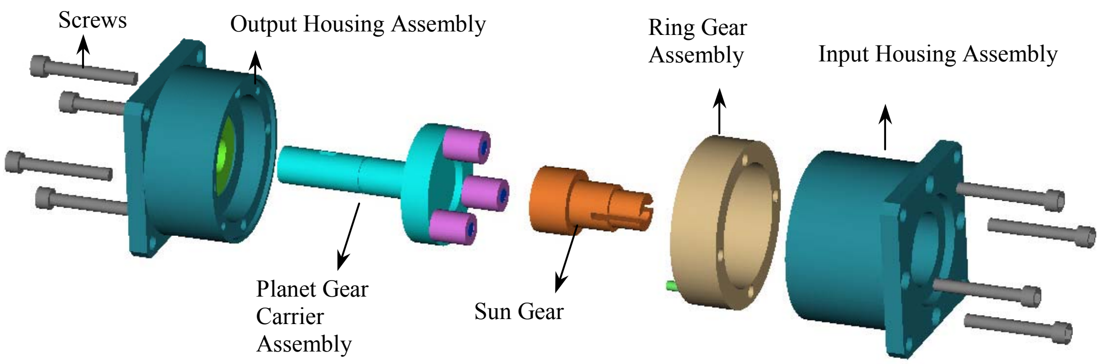


Figure 35: Planetary gear system Assembly

Consider the output housing assembly and planet gear-carrier assembly shown in Figure 36. The output shaft of the planet gear-assembly is inserted into the bearings of the output housing assembly.

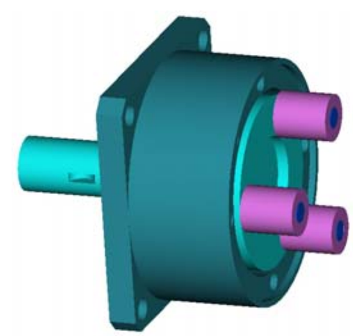


Figure 36: Output Housing Assembly and Planet Gear-Carrier Assembly

Table 6 shows the assembly relationships, and Figure 37 shows the instance diagram of the two artifacts. Note that although there are two assemblies involved, the actual relationships are formed by three parts in the assemblies. The instances of the assemblies are also shown in Figure 37 to illustrate the *part-of* relationships. The artifact association *mc4* involves the three parts of output shaft, bearing 1, and bearing 2. It is instantiated from **MovableConnection** since the output shaft and two bearings constitute a revolute pair. The artifact association *mc4* and the associated assembly feature association are ternary relationships.

Artifact associations	Artifacts	Assembly features	Assembly constraints	Kinematic relationships
mc4	Output shaft @Planet carrier assembly @Planet gear-carrier assembly	Bearing seat of output shaft surface (bearingSeat3:AF)	Coaxial Parallel Angle with dimension	Relative rotation (rp5:RevolutePair)
	Bearing 1 @Output housing assembly	Inner race surface of bearing (innerRace1:AF)		
	Bearing 2 @Output housing assembly	Inner race surface of bearing (innerRace2:AF)		

Table 6: Assembly Relationships Between Output Housing Assembly and Planet Gear-Carrier Assembly

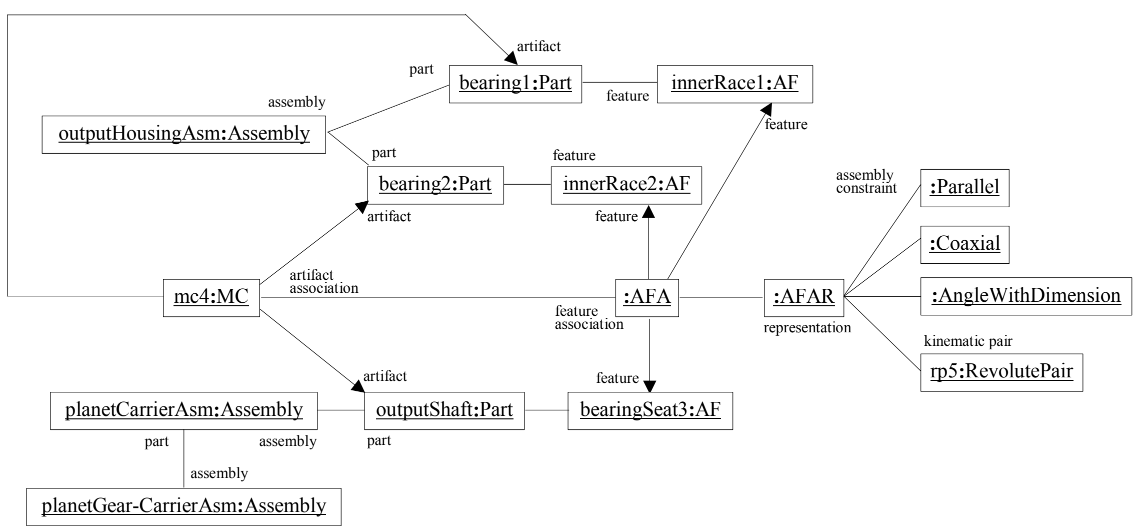


Figure 37: Output Housing Assembly and Planet Gear-Carrier Assembly Instance Diagram

Let us now consider the sungear and planet gear-carrier assembly, shown in Figure 38. The sungear is assembled with the three planet gears of the planet gear-carrier assembly by gear meshing.

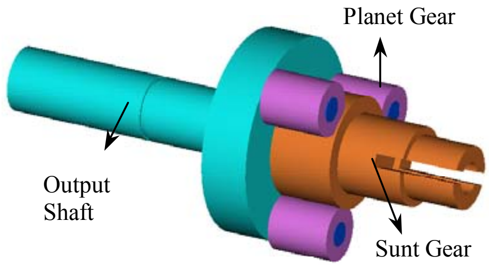


Figure 38: Sungear And Planet Gear-Carrier Assembly

The assembly relationships are listed in Table 7 and the instance diagrams are illustrated in Figure 39. Five parts participate in the current assembly relationships. Three movable connections *mc5*, *mc6*, and *mc7* are instantiates of **MovableConnection**. To describe the details of the gear meshing, three instances of **GearPair**, namely, *gp1*, *gp2*, and *gp3* in Figure 39 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 7 and Figure 39. 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 39. On the other hand, to position the sungear, parametric assembly constrains need to be assigned.

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 39). 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 39.

Artifact associations	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:RevolutePair)

Table 7: Assembly Relationships Between Sungear and Planet Gear-Carrier Assembly

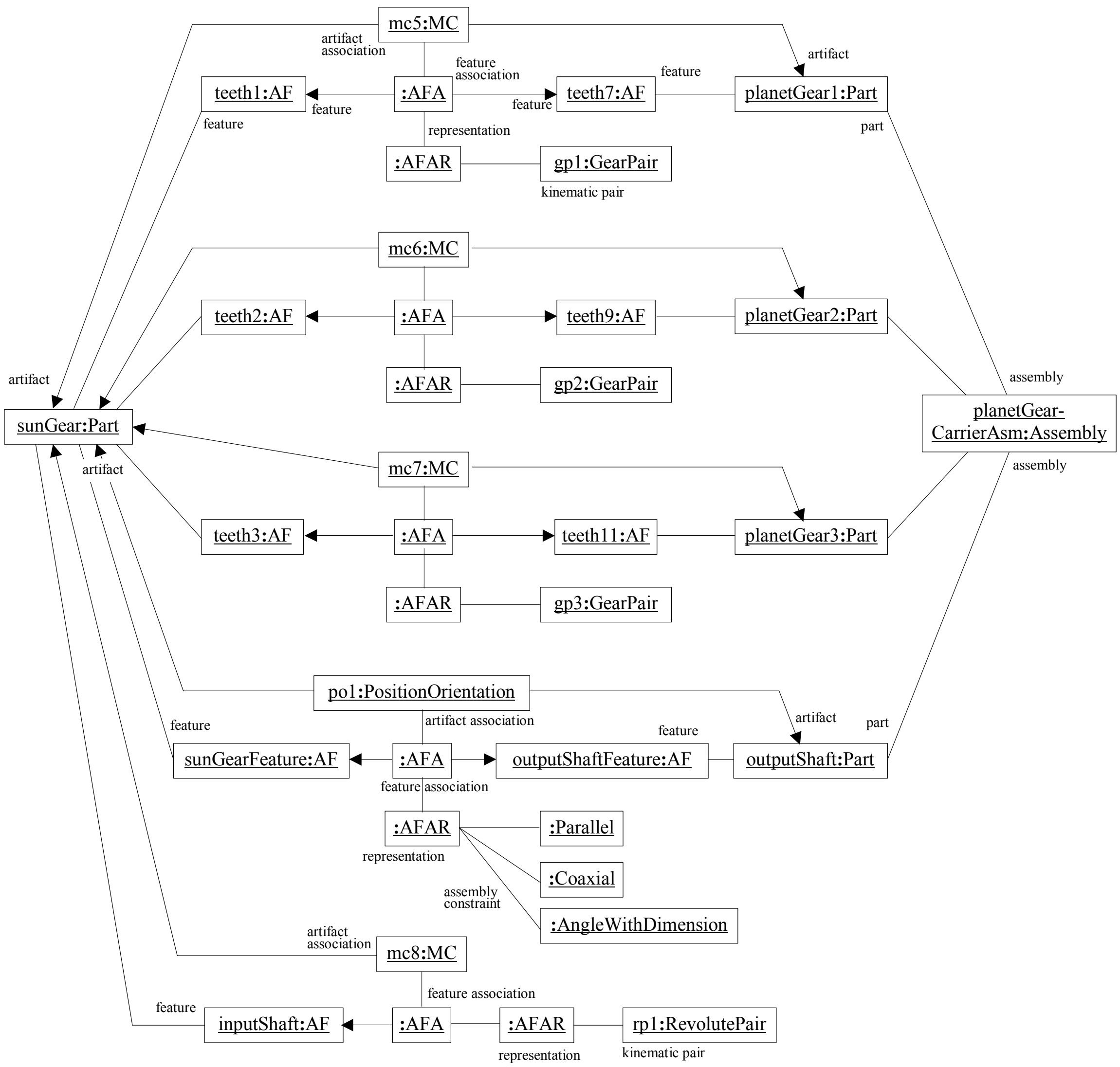


Figure 39: Sungear and Planet Gear-Carrier Assembly-Instance Diagram

Next, let us consider the planet gear-carrier assembly and ring gear assembly shown in Figure 40. The ring gear is meshed with the three planet gears of the planet gear-carrier assembly.

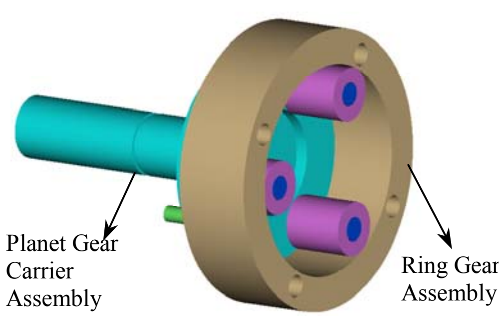


Figure 40: Ring Gear Assembly and Planet Gear-Carrier Assembly

The assembly relationships are shown in Table 8, and Figure 41 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**.

Artifact associations	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 8: Assembly Relationships Between Ring Gear Assembly and Planet Gear-Carrier Assembly

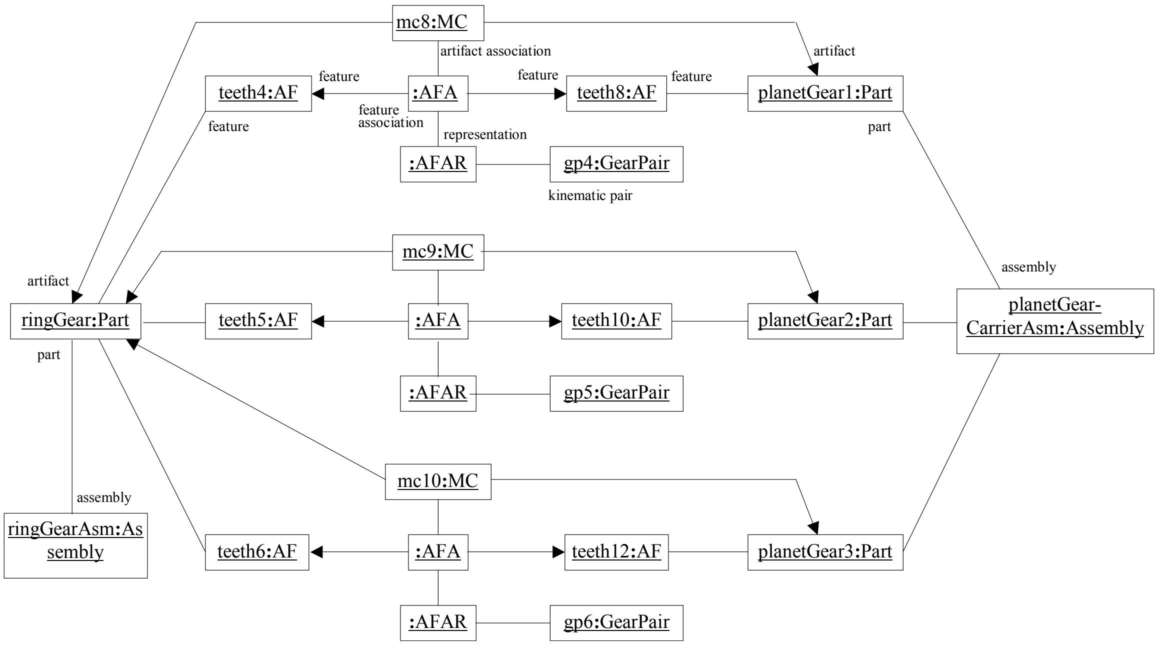


Figure 41: Ring Gear Assembly and Planet Gear-Carrier Assembly-Instance Diagram

Next we model the ring gear assembly, output housing assembly, and screws shown in Figure 42. The ring gear assembly is aligned with the output housing assembly by ring-gear pin and pinhole assembly features. Four screws are used to fasten the output housing assembly into the ring gear assembly.

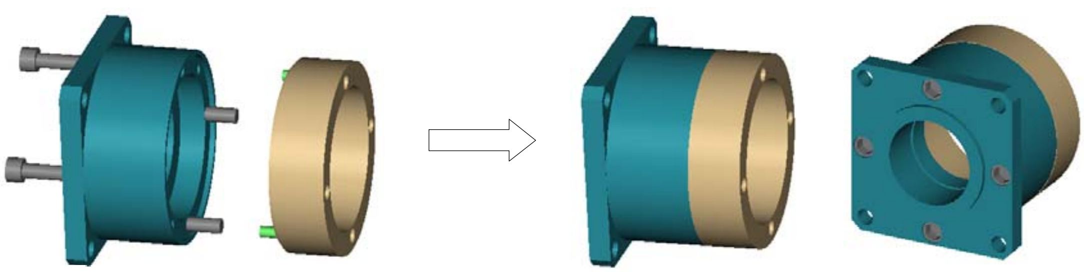


Figure 42: Output Housing Assembly, Ring Gear Assembly, and Screws

Table 9 shows the assembly relationships. Figure 43 shows the instance diagrams. First, the fixed connection *fc9* shows the assembly relationships between the two ring-gear pins and the output housing. The associated assembly feature association involves four assembly features: the two pinhole features (pinHole9 and pinHole10) are from the output housing and the other two features (pinCylinder9 and pinCylinder10) are from the two ring-gear pins. To this assembly feature association, the parametric assembly constraints are attached to position the ring gear to the output housing, as shown in Figure

43. Second, there is another fixed connection, *fc10*, which represents the fastening relationship by four screws. It involves six artifacts: output housing, ring gear, and four screws. In addition, it has four assembly feature associations, each of which relates the assembly features, as shown in Figure 43.

Artifact associations	**Artifacts**	**Assembly features**	**Assembly constraints**	**Kinematic relationships**
fc10	Ring-gear pin1 @Ring gear assembly	Inserted portion of pin surface (pinCylinder9:AF)	Coaxial Coaxial Parallel	No relative motion
	Ring-gear pin2 @Ring gear assembly	Inserted portion of pin surface (pinCylinder10:AF)		
	Output housing @Output housing assembly	Two pin-hole surfaces (pinHole9~10:AF)		
fc11	Output housing @Output housing assembly	Four through holes (thruHole1~4:AF)	Coaxial Angle with dimension Parallel (for each screw)	No relative motion
	Ring gear @Ring gear assembly	Four threaded holes (threadedHole1~4:AF)		
	Screw1 - Screw4	Four thread shanks of screws (thread1~4:AF)		

Table 9: Assembly Relationships Between Output Housing Assembly, Ring Gear Assembly, and Screws

Finally, consider the input housing and ring gear assembly shown in Figure 44. The input housing is assembled with the ring gear assembly. Four screws are used to fasten the input housing into the ring gear assembly

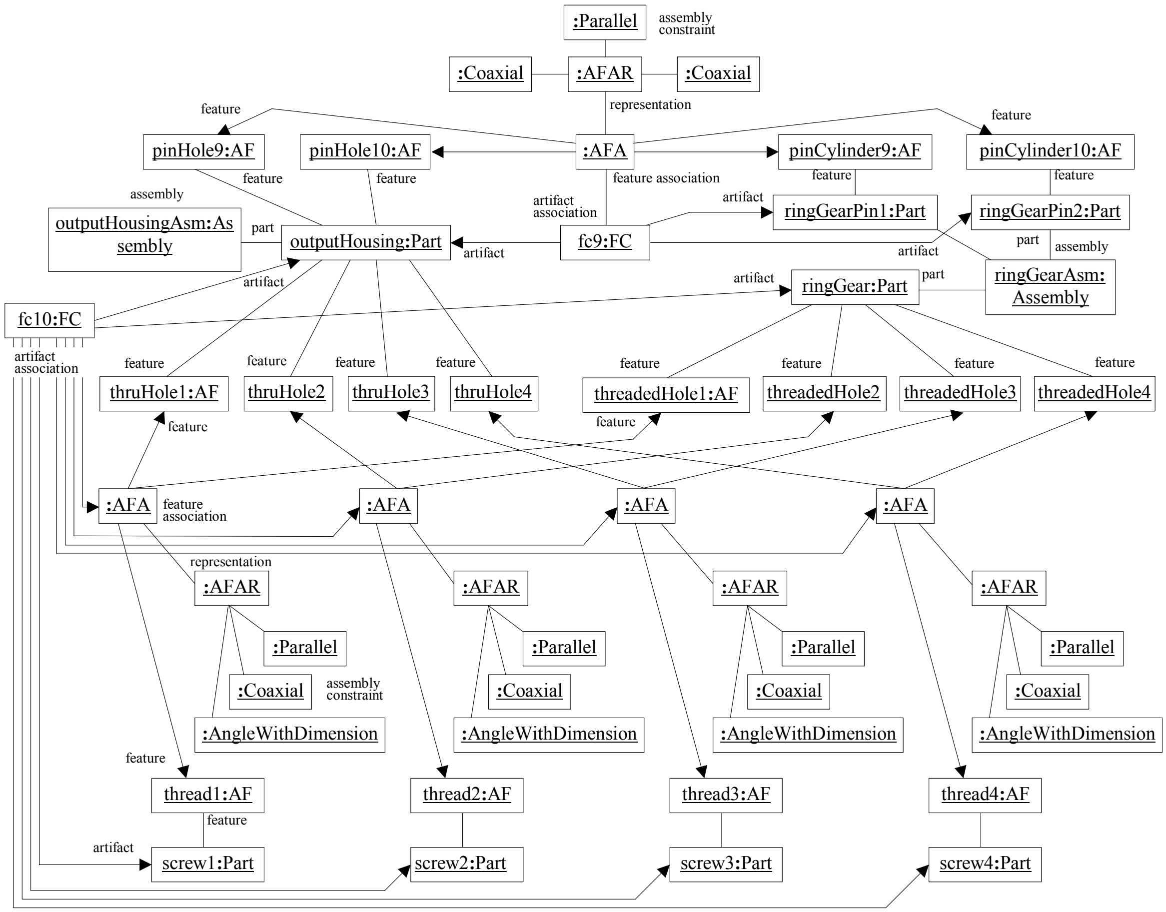


Figure 43: Output Housing Assembly, Ring Gear Assembly, and Screws-Instance Diagram

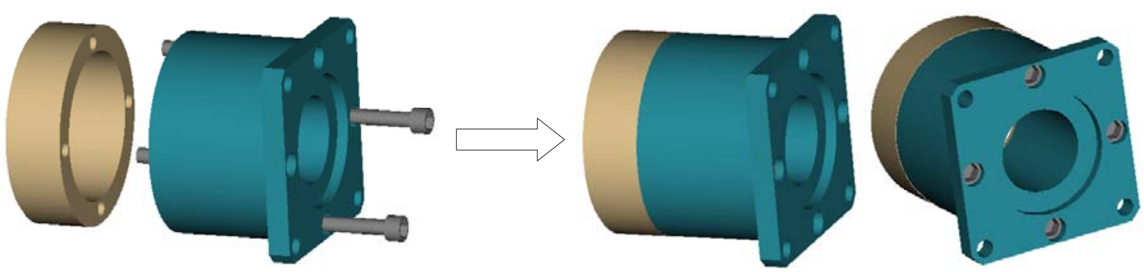


Figure 44: Input Housing, Ring Gear Assembly, and Screws

Table 10 lists the assembly relationships. Figure 45 shows the instance diagram. The fixed connection *fc11* represents the assembly relationships between the input housing and the ring gear of the ring gear assembly. The assembly feature association attached to *fc11* defines the parametric assembly constraints to position the input housing with respect to the ring gear. The fixed connection *fc12* has a similar structure to that of *fc10* as explained previously. It represents the fastening of the input housing and ring gear assembly by four screws.

Artifact associations	Artifacts	Assembly features	Assembly constraints	Kinematic relationships
fc12	Ring gear @Ring gear assembly	Input-housing side of ring gear surface (ringGearSide:AF)	Coaxial Coaxial Parallel	No relative motion
	Input housing	Stepped side (steppedSide:AF)		
fc13	Ring gear @Ring gear assembly	Four threaded holes (threadedHole1~4:AF)	Coaxial Angle with dimension Parallel (for each screw)	No relative motion
	Input housing	Four through holes (thruHole5~8:AF)		
	Screw5 – Screw8	Four thread shanks of screws (thread5~6:AF)		

Table 10: Assembly Relationships Between Input Housing, Ring Gear Assembly, and Screws

4.4 Kinematic Information Representation

Figure 46 shows the kinematic diagram of the planetary gear system, where only the parts (kinematic links) involved in motion (or force) transmission are illustrated. The input motion is transmitted into the sungear through the input shaft that is part of the sungear. Three planet gears are meshed with the sungear, and with the fixed ring gear. The planet carrier assembly (i.e. output shaft + planet-gear pins) holds the three planet gears and rotates with the output shaft that is part of the planet carrier. The output shaft is connected with a bearing. For the input shaft, we assume that it is sustained by an unknown support (or ground).

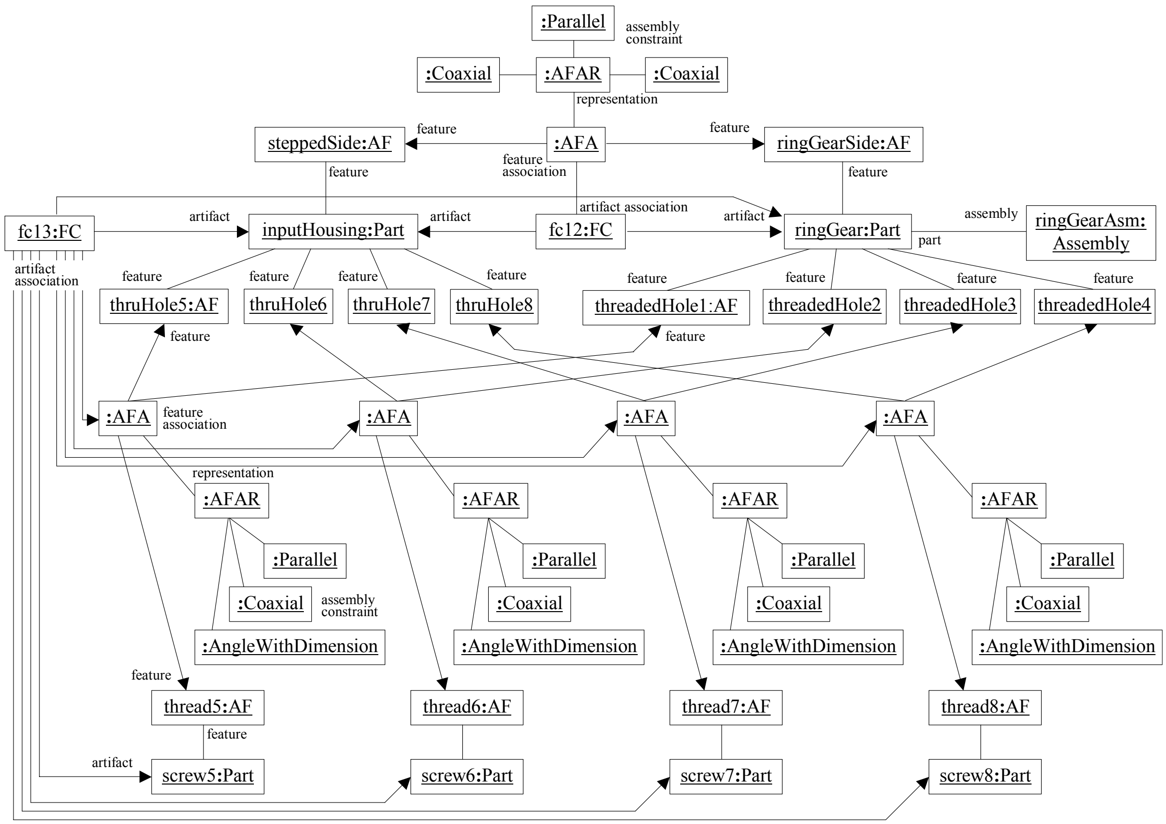


Figure 45: Input Housing, Ring Gear Assembly, and Screws-Instance Diagram

The gear system is composed of two kinematic loops. The first loops comprises the support, the sungear, the planet gear(s), the ring gear, and the bearing. The second involves the planet carrier, the planet gear(s), and the ring gear. The two separate loops are coupled over a common kinematic pair: the planet gears-ring gear pair. The support is actually unknown in this example: the input shaft may be supported by the bearing or connected directly to the output shaft of an electric motor or other components. In any case, the input shaft should be supported by a structure, and thus an unknown support is presumed in this example, constituting a revolute pair with the input shaft (sungear). The input and output shafts are rigidly attached to the sungear and planet carrier, respectively. The support, ring gear, and bearing are assumed grounded (fixed).

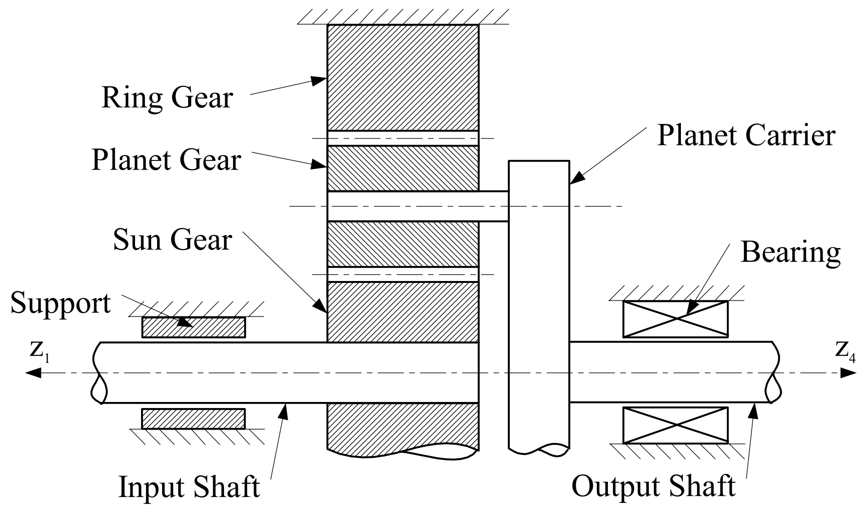


Figure 46: Kinematic Diagram of Planetary Gear System

Table 11 shows the kinematic pairs and the associated parts that are identified from the planetary gear system. For convenience, numbers are used to distinguish the three planet gears and the kinematic pairs of the same type. As shown in Table 11, two types of kinematic pairs (**GearPair** and **RevolutePair**) are used in the planetary gear system.

Kinematic Pairs	**Associated Parts**
Revolute Pair 1	Unknown Support – Sungear (Input Shaft)
Gear Pair 1	Sungear – Planet Gear 1
Gear Pair 2	Sungear – Planet Gear 2
Gear Pair 3	Sungear – Planet Gear 3
Gear Pair 4	Planet Gear 1 – Ring Gear
Gear Pair 5	Planet Gear 2 – Ring Gear
Gear Pair 6	Planet Gear 3 – Ring Gear
Revolute Pair 2	Planet Gear 1 – Planet Carrier
Revolute Pair 3	Planet Gear 2 – Planet Carrier
Revolute Pair 4	Planet Gear 3 – Planet Carrier
Revolute Pair 5	Planet Carrier – Bearing

Table 11: Kinematic Pairs and Associated Parts (Links) of Planet Gear System

Before proceeding further with the foregoing example, we should first assign necessary frames to each link (part) of the gear system. In general, two coordinate systems, or frames, are needed to describe the kinematic behavior of any kinematic pair, each attached to a link of the pair. Considering that a binary link is associated with two kinematic pairs (one with the preceding link and the other with the following link), there are two frames associated with a link. If any link has more kinematic pairs than two, as many frames are necessary as the number of involved kinematic pairs.

The assignment of frames is actually arbitrary and depends on individual applications. In this example, assume that the necessary frames are assigned as shown in **Figure 47**. **Figure 47**(a) and (b) are depicted from positive z_1 and z_4 axes, respectively. Each view shows each of the two kinematic loops of the mechanism, and includes the links involved in each loop. Since the mechanisms in the planetary gear system are planar mechanisms, and the coordinate systems need not be attached directly to each link, the coordinate systems can be thought of as being on any one plane in this example.

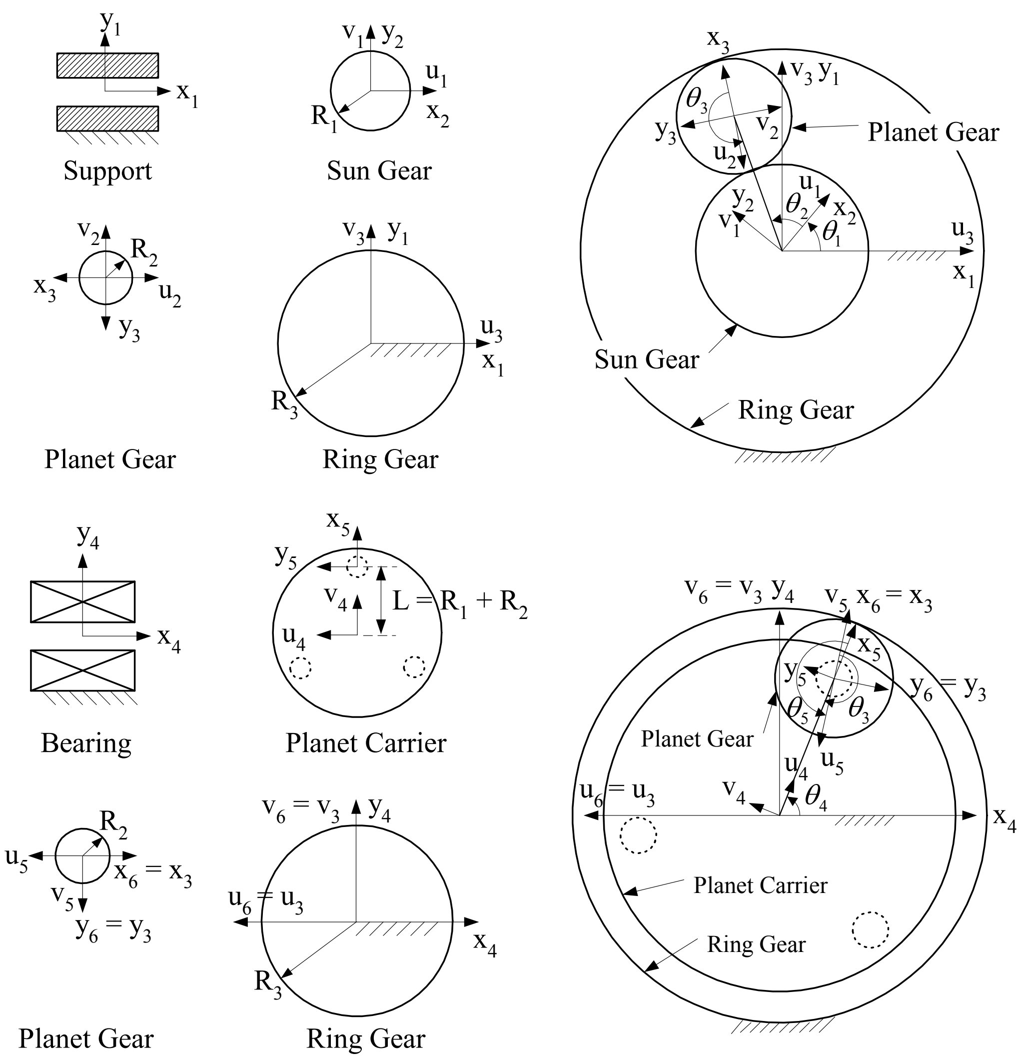


Figure 47 (a), (b): Coordinate Systems Assigned To Kinematic Pairs

Table 12 shows the frames and the pair variables of each kinematic pair of the planetary gear system. The frame $\{x_i y_i z_i\}$ is attached to the preceding link of the pair, and $\{u_j v_j w_j\}$ to the following link. A *pair variable* is a variable parameter of the SU-parameters (see

Appendix A and Reference [18]) defining a kinematic pair. Except the pair variable, other parameters have fixed values. For a revolute pair, the pair variable is defined by the angle between x_i and u_i. In the case of a gear pair, the pair variable is given by the angle from x_i to a common perpendicular between the two frames involved. In Table 12, only one planet gear (Planet Gear 1) and its associated pairs are assigned the necessary coordinate systems for brevity as the three planetary gear are configured symmetrically in space and they play a similar role in the mechanism. The frames associated with the other planet gears are therefore not presented specifically and just denoted as '…' in Table 12. In fact, they are not necessary for the analysis of the gear system due to the symmetry.

Kinematic Pairs	**Associated Parts**	**Frames**	**Pair Variables**
Revolute Pair 1	Unknown Support – Sungear (Input Shaft)	$\{x_1y_1z_1\}$–$\{u_1v_1w_1\}$	θ_1
Gear Pair 1	Sungear – Planet Gear 1	$\{x_2y_2z_2\}$–$\{u_2v_2w_2\}$	θ_2
Gear Pair 2	Sungear – Planet Gear 2	…	…
Gear Pair 3	Sungear – Planet Gear 3	…	…
Gear Pair 4	Planet Gear 1 – Ring Gear	$\{x_3y_3z_3\}$–$\{u_3v_3w_3\}$	θ_3
Gear Pair 5	Planet Gear 2 – Ring Gear	…	…
Gear Pair 6	Planet Gear 3 – Ring Gear	…	…
Revolute Pair 2	Planet Carrier – Planet Gear 1	$\{x_5y_5z_5\}$–$\{u_5v_5w_5\}$	θ_5
Revolute Pair 3	Planet Carrier – Planet Gear 2	…	…
Revolute Pair 4	Planet Carrier – Planet Gear 3	…	…
Revolute Pair 5	Bearing – Planet Carrier	$\{x_4y_4z_4\}$–$\{u_4v_4w_4\}$	θ_4

Table 12: Frames of Each Kinematic Pair

The total number of associated frames on a link depends on the type and usage of the link, as shown in Table 13. The frame $\{u_iv_iw_i\}$ is related with the preceding pair of a link (in conjunction with the previous link), and $\{x_iy_iz_i\}$ with the following pair (in conjunction with the following link). The unknown support and bearing in Table 13 contains only one frame $\{x_iy_iz_i\}$. These parts are assumed grounded (fixed) links. Hence, they have no associated links preceding them, and thus frame $\{u_iv_iw_i\}$ is not necessary for them. The sungear, ring gear, and planet carrier will in fact have more frames since they are connected to three planet gears. That is, they are quaternary links and will have four frames in all. However, their configuration in this example is symmetric and thus the frames of the other pairs are left out for brevity. In the case of the ring gear, there are three frames listed. The frame pair $\{u_3v_3w_3\}$ and $\{x_1y_1z_1\}$ is used in the first loop (**Figure 47**(a)), and frame pair $\{u_3v_3w_3\}$ and $\{x_4y_4z_4\}$ in the second loop (**Figure 47**(b)). The frames $\{x_1y_1z_1\}$ and $\{x_4y_4z_4\}$ represent the same ground frames to which the ring gear is fixed. The reason that two different frames are introduced is for convenience in assigning the frames. We can use the same frame for the ground, if necessary. The planet gears are ternary links. They are respectively connected to sungear, ring gear, and planet carrier, respectively. Thus, there exist three frames for a planet gear, as shown in Table 13.

Parts	Frames
Unknown Support (Ground)	$\{x_1y_1z_1\}$
Sungear	$\{u_1v_1w_1\}$, $\{x_2y_2z_2\}$, ...
Planet Gear 1 (2, or 3)	$\{u_2v_2w_2\}$, $\{u_5v_5w_5\}$, $\{x_3y_3z_3\}$
Ring Gear	$\{u_3v_3w_3\}$, $\{x_1y_1z_1\}$, $\{x_4y_4z_4\}$, ...
Planet Carrier	$\{u_4v_4w_4\}$, $\{x_5y_5z_5\}$, ...
Bearing (Ground)	$\{x_4y_4z_4\}$

Table 13: Frames Associated with Each Link (Part)

In the previous subsection 4.3.5, the kinematic pairs were illustrated in the instance diagrams of the planetary gear assembly. To demonstrate the kinematic relationships in more detail, another instance diagram is depicted in Figure 48. When the instances are of the same class, only the first occurrence of them is denoted with the class name. This diagram includes all the parts or assemblies, and other information that are related to the kinematics of the planetary gear system. Notice in Figure 48 that the three planet gear pins and the output shaft are attached by fixed connections (see Section 4.3.3). Thus they are considered to be one rigid body (we name it planet carrier) for the purpose of kinematic analysis in this section.

The first row in Figure 48 shows the individual parts or assemblies (instances of **Part** or **Assembly**) that take part in the kinematic relationship. Each part (or assembly) has its own assembly features that are defined to represent the shape aspects of the kinematic pairs. A cylinder/hole-type pair can be defined as assembly features for a revolute pair, and a teeth/teeth-type pair for a gear pair. For example, there are four assembly features for the sungear (ss: cylinder type; *t1*, *t2*, and *t3*: teeth type). The cylinder type is used to represent the revolute pair with the unknown support, and the three teeth types are respectively used to represent the gear pairs with the three planet gears around the sungear.

The assembly feature association captures the topological relationship of kinematic pairs

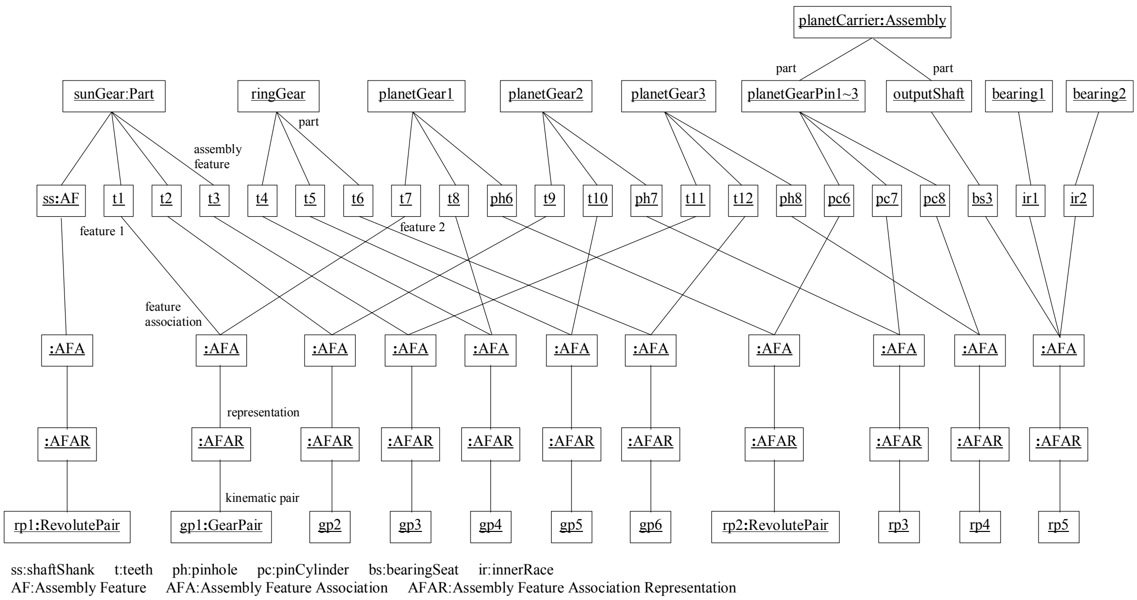


Figure 48: Instance Diagram of Planetary Gear System

(AFAs in Figure 48). First, it associates two (or more when necessary) assembly features that take part in an assembly relationship (specifically, kinematic relationship). By consulting the ownership of an assembly feature, it can finally specify the links (parts) that take part in the relationship. For example, the second instance of assembly feature association in Figure 48 relates assembly features *t1* and *t7* to represent the gear-pair relationship between the sungear and the mating planet gear (planetGear1). Second, it refers to an assembly feature association representation (AFAR in Figure 48), which describes the details of the relationship. In this kinematic example, the relationship to be represented is a kinematic pair (**GearPair** or **RevolutePair).** For instance, the second instance of assembly feature association in Figure 48 is associated with assembly feature association representation, by which the kinematic pair gp1:GearPair is associated (see Figure 48).

The kinematic pairs (instances of **GearPair** and **RevolutePair** in Figure 48) contain their specific kinematic information (constraints of the pair) to describe their own behavior. For example, the instance diagram about rp1:RevolutePair (unknown support – sungear pair) in Figure 48 is illustrated in detail in Figure 49. An instance of **RevolutePairRange** specifies the lower and upper bounds of the rp1:RevolutePair. The current value of the pair variable θ_1 is specified by an instance of **RevolutePairValue**. The two frames required for this revolute pair are provided by two instances of **PairFrame**. Typically, the pair variable θ_1 is measured by the angle from x_1 to u_1, as shown in **Figure 47**(a).

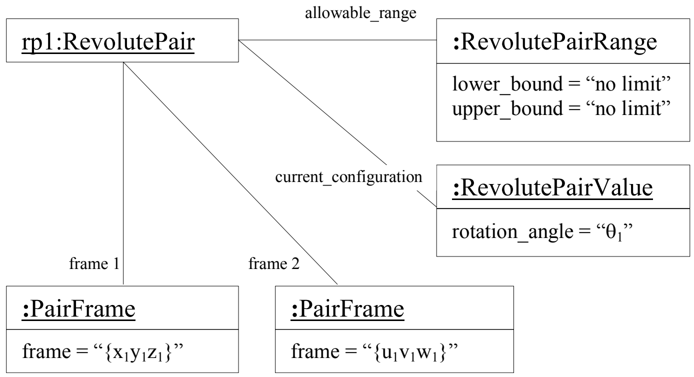


Figure 49: Instance Diagram of RevolutePair (RP1: RevolutePair)

As another example, the instance diagram for gp1:GearPair is shown in Figure 50. The gp1:GearPair contains some key geometric information of the gears such as radii, gear ratio, bevel and helical angles (number 1 denotes the first link, and number 2 the second). Such information is necessary when deriving the kinematic equation of the gear pair. An instance of **GearPairRange** sets limits on the range of gears' rotation angle. The current rotation angle of the first gear is specified by θ_2 of an instance of **GearPairValue**; the rotation angle of the second gear can be derived from the rotation angle of the first gear by multiplying it with the gear ratio.

Instance diagrams for the other kinematic pairs can also be drawn in a similar manner. As mentioned earlier, *gp2* and *gp3* will have the same structure as *gp1* since they are symmetrically configured with *gp1*. The gear-pair instances *gp4, gp5*, and *gp6* will also have similar instance diagram to that of *gp1* (see Figure 50), except that they are internal gearing: their gear ratios are negative values. For other revolute pairs *rp2, rp3, rp4*, and *rp5*, instance diagram similar to that of *rp1* (see Figure 49) can be depicted.

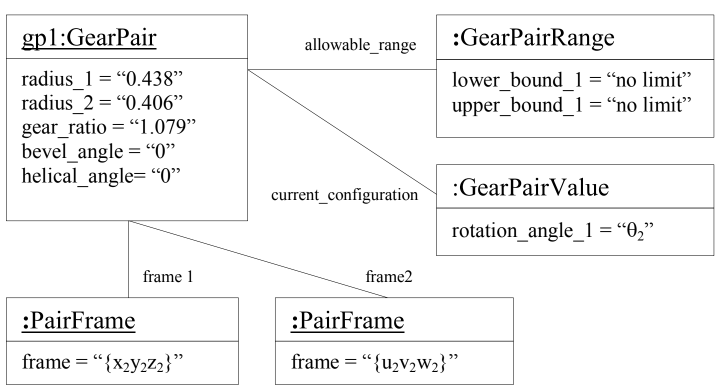


Figure 50: Instance Diagram of GearPair (GP1: GearPair)

4.5 Tolerance Chains In The Planetary Gear System Design

Tolerance chains are important in a product assembly. The chains are used in the analysis for assembleability and functions such as clearance, tightness, smooth motion, and flow rate. This subsection describes the modeling of tolerance chains. Two specific tolerance chains are found in the planetary gear system assembly. The first one is in the axial direction. Three tolerances are in this chain. They are all defined on the part of the sungear and the input shaft. This chain of the tolerances determines the magnitude of clearance or interference between the Surface B of the sungear and the Surface A on the planetary gear holder, as shown in Figure 51. (Note that Surface B is the end surface of the sungear). Figure 52 shows the chain of the three dimensions and the associated tolerances: 12.95 mm ± 0.01 mm on the sungear, 13.59 mm ± 0.03 mm on the shank, and 10.8 mm ± 0.1 mm on the connector. The total allowable variation of the sungear's length is, therefore, 0.14 mm. These three dimensional tolerances can be represented using objects derived from the **DimensionalTolerance** class described in Section 3.11. Note that not all the dimensions of this part are shown in Figure 52. Only those dimensions related to the tolerance chain are shown.

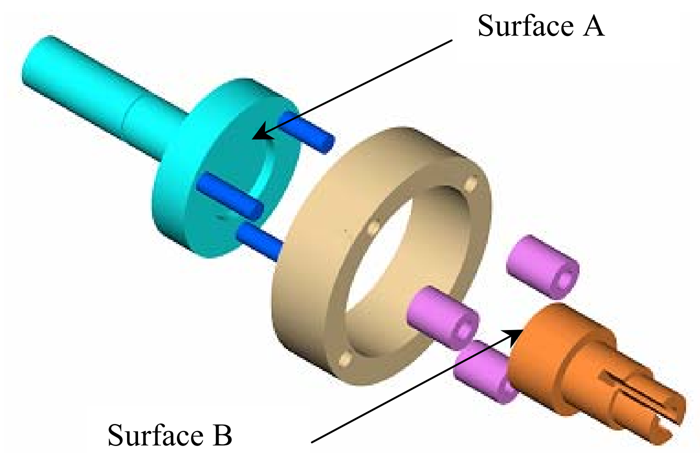


Figure 51: The assembly relationship between the sungear and the planetary gear holder

Figure 52 shows an engineering drawing of the sungear. The tolerance instance diagram (Figure 53) shows how the tolerances are represented using UML as discussed in Section 3.11. Figure 53 shows an integrated view of the part, its tolerances, and the assembly features.

In addition to dimensional tolerances, geometric tolerances are also used to control three features of this part. The end surface (Surface B) has two tolerances: a perpendicularity tolerance and a flatness tolerance. Both are for the control of geometric variation of Surface B. The perpendicularity tolerance is used to control the wobbling motion of the surface while the gear is rotating around the datum axis (Datum A1). The flatness tolerance is used to control the proper clearance between Surface B and Surface A on the planetary gear holder. A cylindricity tolerance is applied to the surface of the shank. It is used to control the form variation of the surface. These three tolerances are form tolerances. A fourth tolerance is the coaxiality tolerance. The coaxiality of the connector is controlled by a position tolerance. The detail of coaxiality tolerancing can be found in the dimensioning and tolerancing standard [19]. The three form tolerances are instances of **PerpendicularityTolerance, FlatnessTolerance,** and **CylindricityTolerance** classes respectively, described in Section 3.11.1. The position tolerance is an instance of the **PositionTolerance** class described in Section 3.11.3.

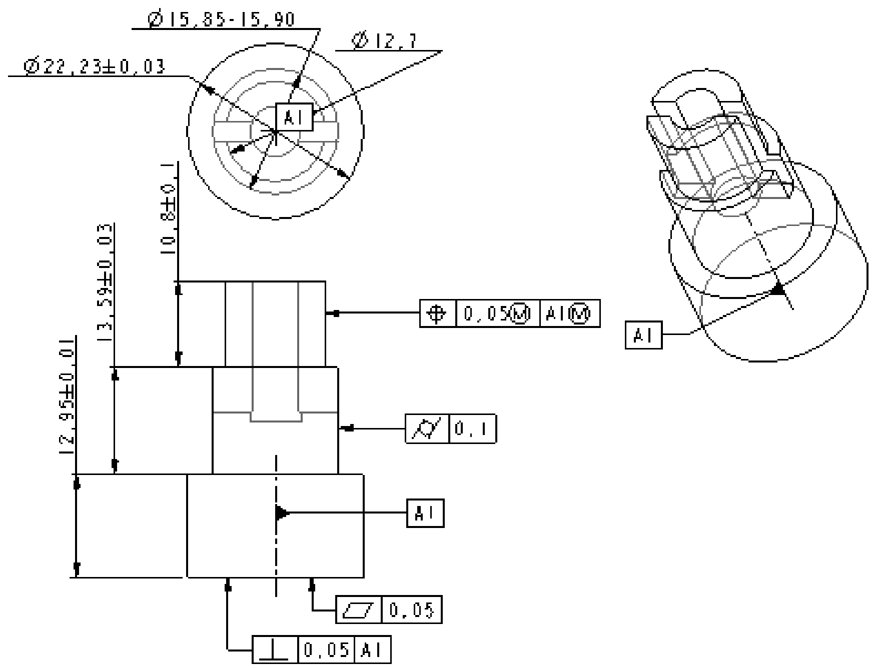


Figure 52: Sungear Tolerances

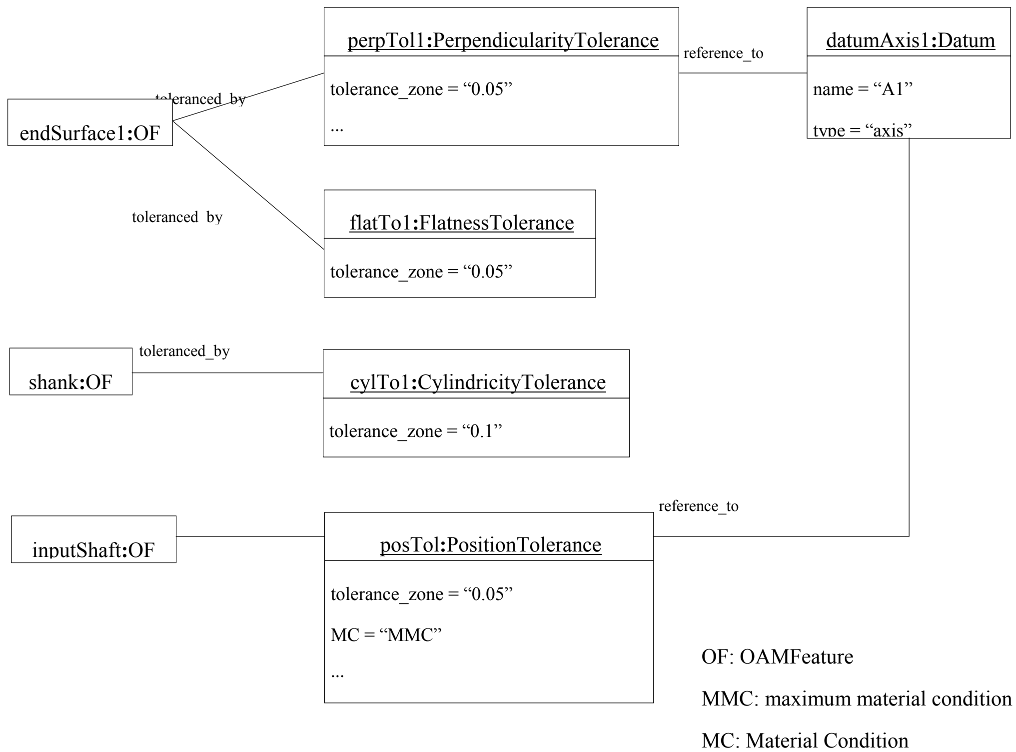


Figure 53: Tolerance Instance Diagram

The second tolerance chain is in the radial direction of the planetary gear system. This chain includes three parts: the sungear, the planetary gears, and the ring gear. This chain starts from the axis of the sungear. The pitch diameter of the gear is 22.23 mm ± 0.03 mm, as shown in Figure 52.

The second set next to the sungear in the tolerance chain is the set of three planetary gears. Figure 56 shows the engineering drawing of a planetary gear. The tolerance instance diagram (Figure 55) shows how the tolerances are represented using UML. Figure 57 shows an integrated view of the part, its tolerances, and the assembly features. Table 14 tabulates the features, tolerances, and control frames with the artifact association.

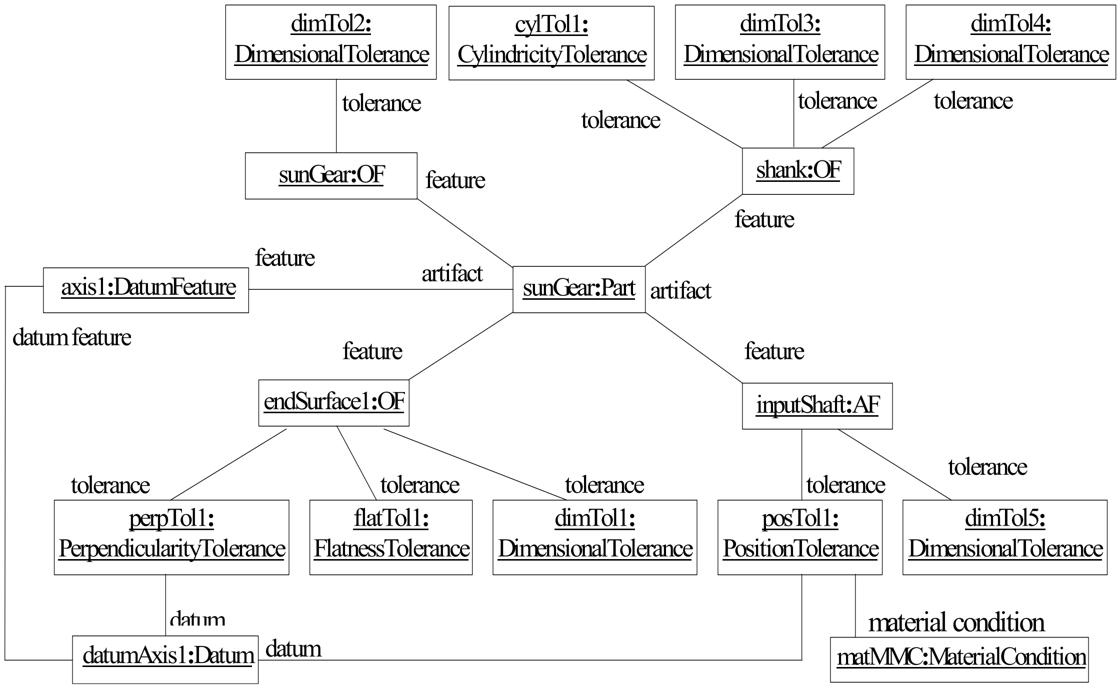


Figure 54 Sungear Part Instance Diagram

The pitch diameter of the planetary gear is 10.16 mm ± 0.01 mm. The planetary gears have the same tolerance and are assembled with the sungear in the radial direction. There is a form tolerance applied to the planetary gear. A cylindricity tolerance of 0.02 mm controls the inner cylindrical surface of the gear. It is used to ensure a tight fit of the gear and the pin, as shown in Figure 51.

Features	**Tolerances**	**Feature Control Frames or dimensional tolerance**	**Artifact Association**
Axis (Datum A1) (axis1:DatumFeature)	None	None	None
End surface (endSurface:OF)	Perpendicularity tolerance	⊥ | 0.05 | A1 (perpTol:PerpendicularityTolerance)	None
	Flatness tolerance	⏥ | 0.05 (flatTol:FlatnessTolerance)	
	Dimensional tolerance	(ϕ22.23) ± 0.03 (dimTol1:DimensionalTolerance)	
sungear teeth (sunGearTeeth:OF)	Dimensional tolerance	(12.95) ± 0.01 (dimTol2:DimensionalTolerance)	None
Shank (shank:OF)	Cylindricity tolerance	⌭ | 0.1 (cylTol1:CylindricityTolerance)	None

	Dimensional tolerance	(13.59) ± 0.03 (dimTol3:DimensionalTolerance)	
	Dimensional tolerance	ϕ15.85 ~ ϕ15.90 (dimTol4:DimensionalTolerance)	
Input shaft* (inputShaft:AF)	Position tolerance	⌖ | 0.05 Ⓜ | A1 Ⓜ (posTol1:PositionTolerance)	*mc8*
	Dimensional tolerance	(10.80) ± 0.10 (dimTol5:DimensionalTolerance)	

Table 14: Tolerance Table For The Sungear

(The features postfixed with * are the assembly features that have been used to represent the assembly relationships in previous sections. Their associated artifacts are also shown in the table.)

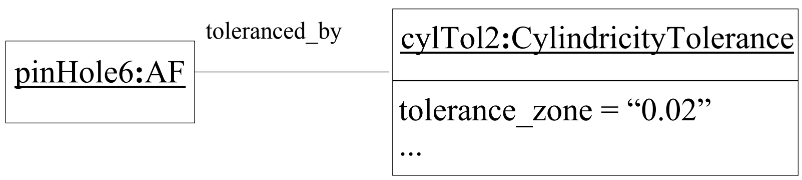


Figure 55: Tolerance Instance Diagram

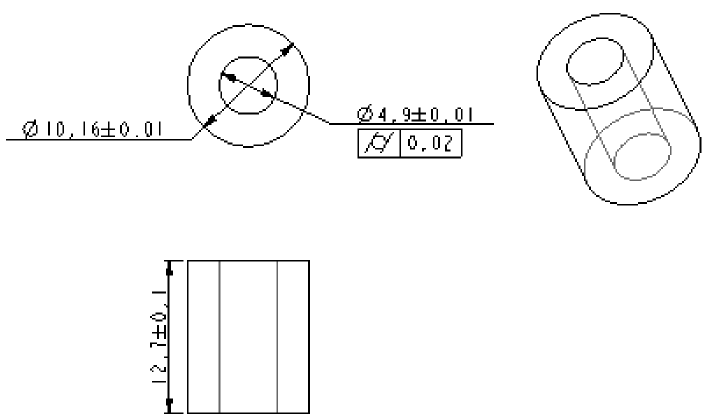


Figure 56: Planetary Gear Tolerances

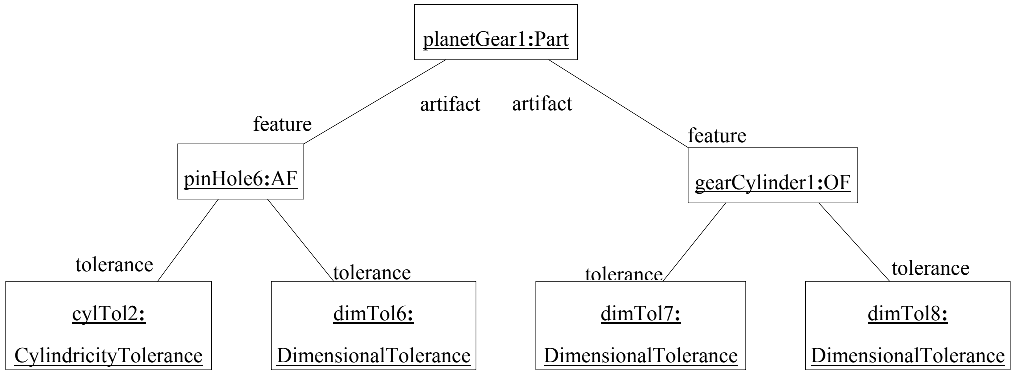


Figure 57: Planetary Gear Part Instance Diagram

Features	**Tolerances**	**Feature Control Frames**	**Artifact Association**
Gear journal hole surfaces* (pinHole6:AF)	Cylindricity tolerance	⌭ 0.02 (cylTol2:CylindricityTolerance)	*mc1~3*
	Dimensional tolerance	(ϕ4.90) ± 0.01 (dimTol6:DimensionalTolerance)	
Gear cylinder (gearCylinder1:OF)	Dimensional tolerance	(12.70) ± 0.10 (dimTol7:DimensionalTolerance)	None
	Dimensional tolerance	(ϕ10.16) ± 0.01 (dimTol8:DimensionalTolerance)	

(Planet gears 2, 3 also have similar tolerances to those of planet gear 1.)

Table 15: Tolerance Table For A Planetary Gear

The third part in the chain is the ring gear, which is assembled with the planetary gears. The pitch diameter of the ring gear is 42.62 mm ± 0.01 mm. Figure 58 shows the engineering design of the ring gear. The tolerance instance diagram (Figure 59) shows how the tolerances are represented using the UML model discussed in Section 3.11.

Figure 60 shows an integrated view of the part, its tolerances, and the assembly features. Table **16** tabulates the features, tolerances, and control frames with the artifact association.

With the dimensional tolerance on the pitch circle of a planetary gear shown in Figure 56 and the dimensional tolerance on the pitch diameter of the sungear, tolerance analysis can be performed in this chain. Examples are clearance or interference of this set of gears. In addition to dimensional tolerances, there is a geometric tolerance on the part. A parallelism tolerance is applied to a surface of the shape of a ring where the ring gear and the output housing meet. Within a tolerance of 0.05 mm, the surface has to be parallel to Datum A3, which is established from the bottom surface of the ring gear. The bottom surface meets with the input housing. This tolerance is to ensure that in the input housing, the ring gear and the output housing are properly aligned along the rotating axes of the gears.

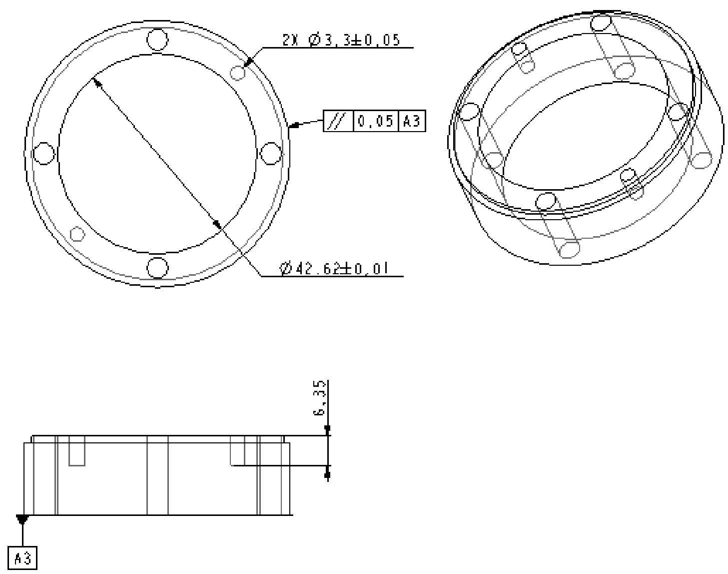


Figure 58: Ring Gear Tolerances

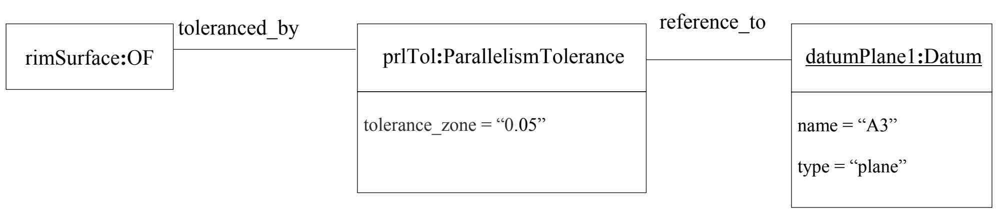


Figure 59: Tolerance Instance Diagram

The chain starts from the axis of the planetary gear system, the same as the axis of the sungear. The radius of the sungear is 11.125 mm and the radial tolerance is, therefore, ± 0.015 mm. The diameter of a planetary gear is 10.16 mm, and its dimensional tolerance is ± 0.01 mm. In the case that both gears are first assembled together, the combined nominal dimension is 21.285 mm (11.125 mm + 10.16 mm), and the combined tolerance is ± 0.025 mm (0.015 mm + 0.01 mm). The nominal radius of the ring gear is 21.31 mm, and the radial tolerance is ± 0.01 mm.

When all three types of gears are assembled, they can be in an extreme tight situation, an extreme loose situation, or any situation in between. In the extreme tight situation, the ring gear radius will be 21.30 mm (21.31 mm – 0.01 mm), and the stack-up is 21.31 mm (21.285 mm + 0.025 mm). In this case, the interference is 0.01 mm. However, in the extreme loose situation, the ring gear radius will be 21.32 mm (21.31 mm + 0.01 mm), and the stack-up will be 21.26 mm (21.285 mm – 0.025 mm). There will be a 0.055 mm clearance among these three kinds of gears.

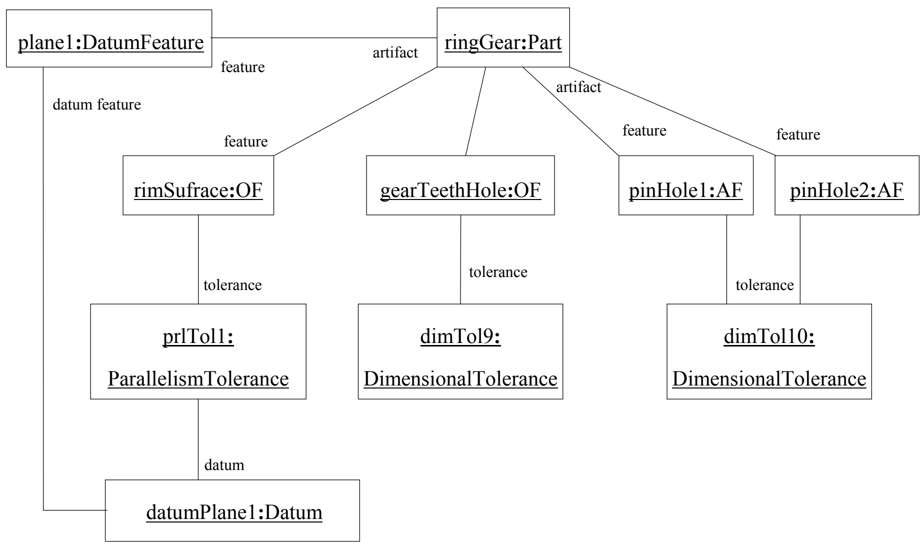


Figure 60: Ring gear part instance diagram

Features	**Tolerances**	**Feature Control Frames**	**Artifact Association**
Plane (Datum A3) (plane1:DatumFeature)	None	None	None
Outer rim surface of boss (rimSurface:OF)	Parallelism tolerance	| // | 0.05 | A3 | (prlTol1:ParallelismTolerance)	None
Inner gear teeth hole (gearTeethHole:OF)	Dimensional tolerance	(ϕ42.62) ± 0.01 (dimTol9:DimensionalTolerance)	None
Two pinhole surfaces* (pinHole1~2: AF)	Dimensional tolerance	(ϕ3.30) ± 0.05 (dimTol10:DimensionalTolerance)	*fc5, fc6*

Table 16: Tolerance table for the ring gear

4.6 Geometric Tolerances In The Planetary Gear System Design

This section describes geometric tolerances in the design. The geometric tolerances are on the three parts: the planetary gear holder, the output housing, and the input housing. Figure 62, Figure 65, and Figure 68 show the drawings of these three parts. Note that not all the dimensions are shown in the drawings. The shown dimensions are primarily applied to assembly and tolerances. Others are shown for showing the proportion of a feature relative to the entire part. Figure 62 shows the engineering drawing of the planetary gear holder. The tolerance instance diagram (Figure 63) shows how the tolerances are represented using the UML model discussed in Section 3.11. Figure 64 shows an integrated view of the part, its tolerances, and the assembly features. Table 17 tabulates the features, tolerances, and control frames with the artifact association.

All the dimensions in this design are with a default tolerance of 0.05 mm, represented by objects derived from the dimensional tolerance class. Datum axis **A** is the rotating axis and can be presented using an object derived from the Datum class defined in Section 3.11. In addition to the datum, three geometric tolerances exist on the drawing. Two geometric tolerances are applied to the end surface adjacent to planetary gears. Since the surface provides support to hold gears, both the instances of **PerpendicularityTolerance** and **FlatnessTolerance** are applied. The perpendicular tolerance makes a reference to Datum A. The third tolerance is an instance of **TotalRunoutTolerance**. It is applied to the output shaft with the tolerance value of 0.1 mm. The runout is with respect to Datum A, the shaft/gear axis, to minimize wobbling motion while the shaft is rotating. A surface profile tolerance is applied to the curved surface of the key way. It is used to ensure the key and the shaft that are tightly fit together.

Figure 65 shows the engineering drawing of the input housing. Figure 66 is```````` the tolerance instance diagram, which shows how the tolerances are represented using the UML model discussed in Section 3.11. Figure 67shows an integrated view of the part, its tolerances, and the assembly features. Table 18 tabulates the features, tolerances, and control frames with the artifact association.

The dimension of the 25.53 mm hole has a statistical tolerance. The statistical tolerance can be represented by an object instantiated from the **StatisticalControl** class in Section 3.11. The mean and standard deviation can be captured using the class described in Section 3.11.6. The variation of the diameter of the counter-bored hole is also statistically controlled, but the depth of the hole is not because it is not a critical dimension. Datum A to be established from the bottom surface is shown in the front view. Datums B and C are shown in the bottom view of the input housing. One position tolerance is applied to a pattern of four holes, each with a counter-bored hole. The position is measured in a reference to a coordinate system established by Datums A, B, and C. The tolerance is in the condition of the regardless of feature size. Another position tolerance is similarly applied to another pattern of four 5.39 mm holes. The positions of these holes are critical in assembly because the input housing and the ring gear have to be properly aligned and, then, assembled

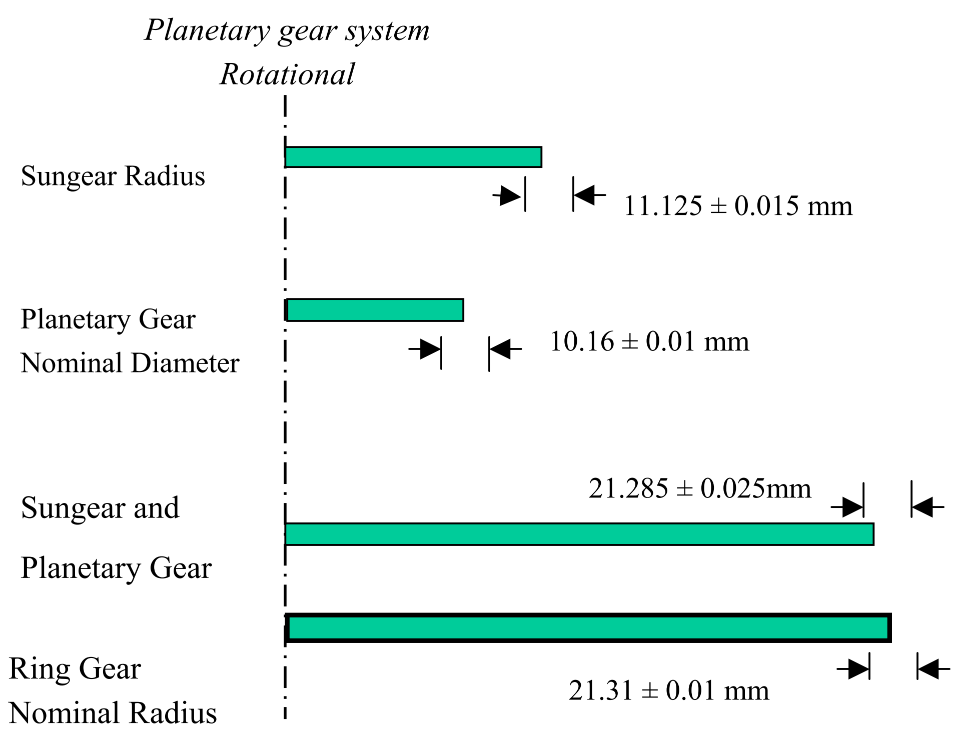


Figure 61: Geometric Tolerances In The Planetary Gear System Design

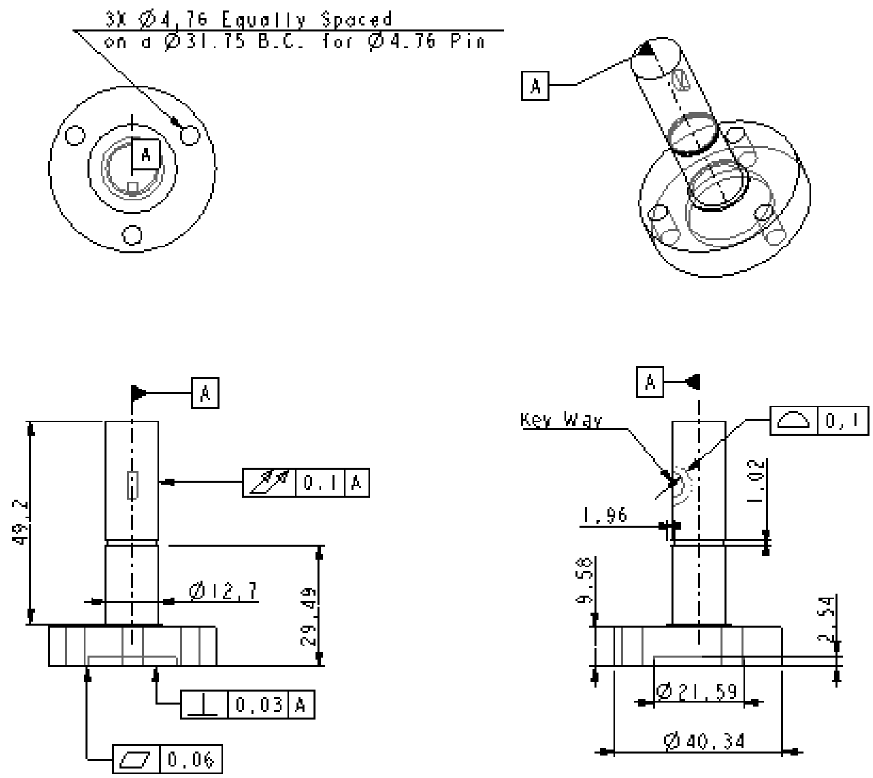


Figure 62: Planetary Gear Holder Tolerances

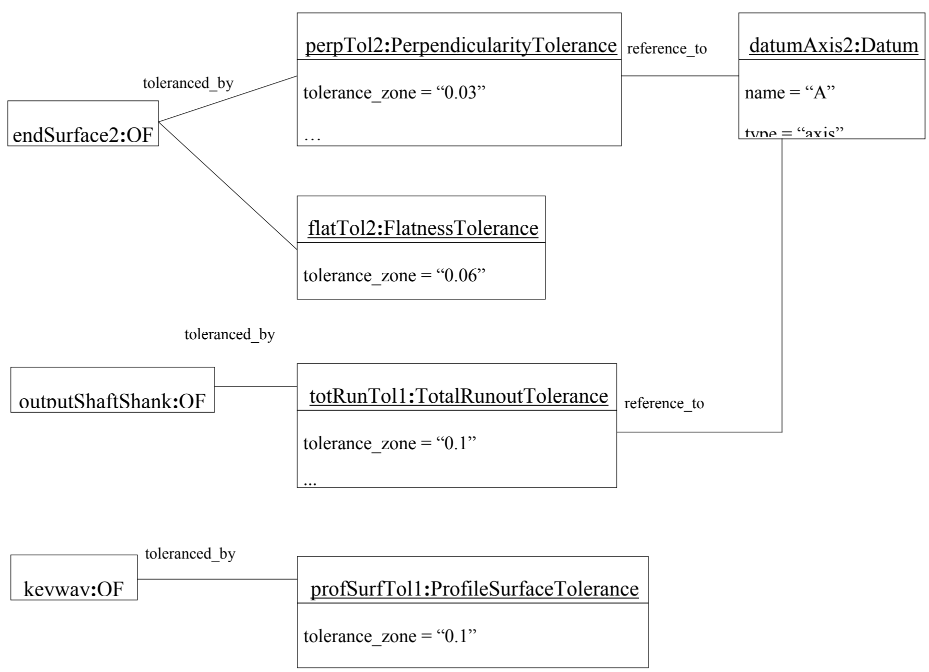


Figure 63: Tolerance Instance Diagram

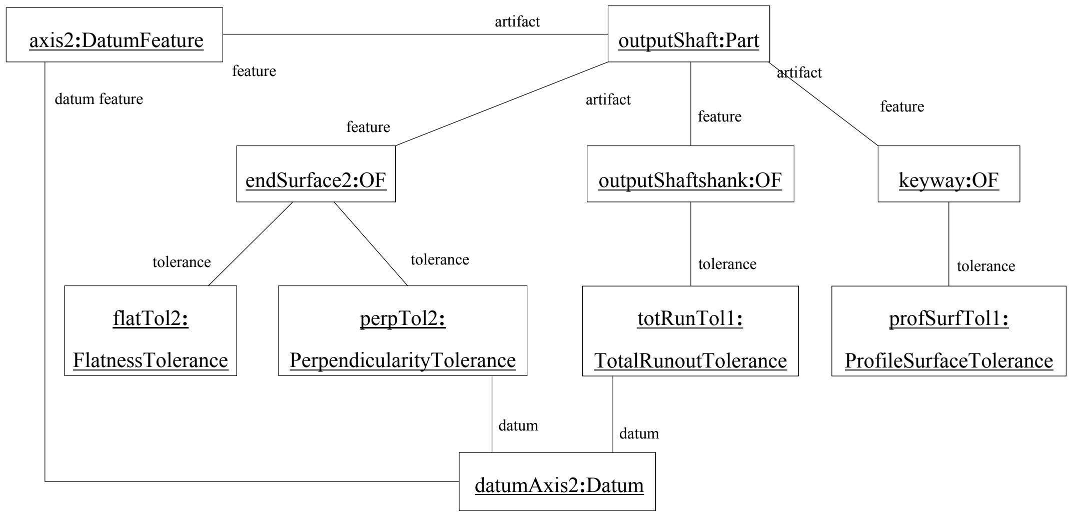


Figure 64: Planetary Gear Holder Part Instance Diagram

Features	**Tolerances**	**Feature Control Frames**	**Artifact Association**
Bottom plane of mounting plate (Datum A) (plane2:DatumFeature)	None	None	None
Side plane of mounting plate (Datum B) (plane3:DatumFeature)	None	None	None
Side plane of mounting plate (Datum C) (plane4:DatumFeature)	None	None	None
Four through holes* (thruHole5~8:AF)	Position tolerance	⌖ | 0.05 | A | B | C (posTol2:PositionTolerance)	*fc13*
Four mounting holes (mountingHole1~4:OF)	Position tolerance	⌖ | 0.05 | A | B | C (posTol2:PositionTolerance)	None
Shaft hole (ShaftHole:OF)	Dimensional tolerance	(ϕ25.53) ± 0.01 (dimTol11:DimensionalTolerance)	None
Counterbore hole (CounterboreHole:OF)	Dimensional tolerance	ϕ37.64 ~ ϕ38.23 ⟨ST⟩ (dimTol12:DimensionalTolerance)	None

Table 17: Tolerance Table For The Planetary Gear Holder

.

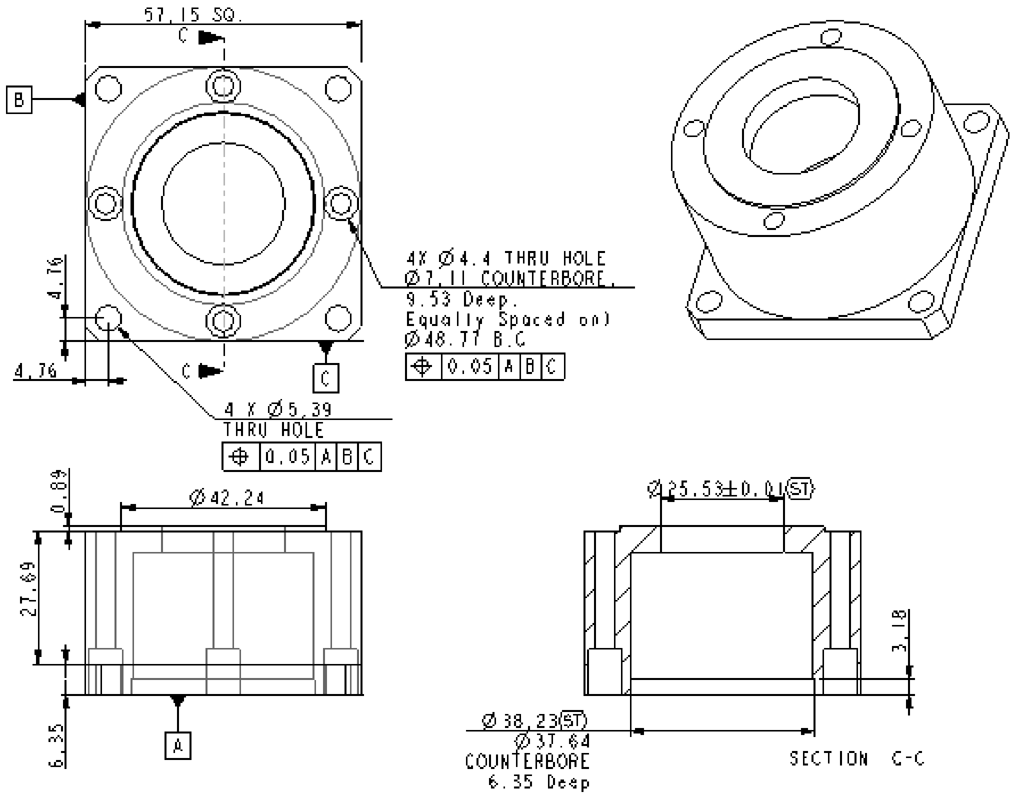


Figure 65: Input Housing Tolerances

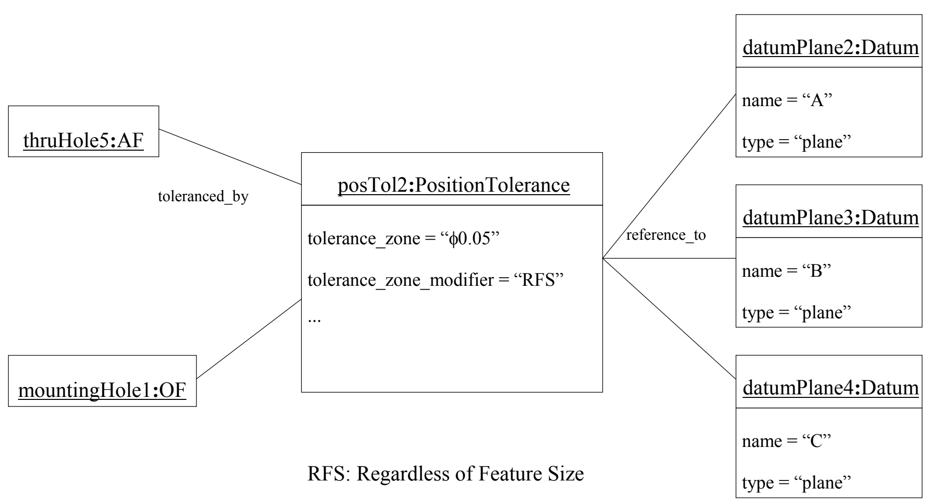


Figure 66: Tolerance Instance Diagram

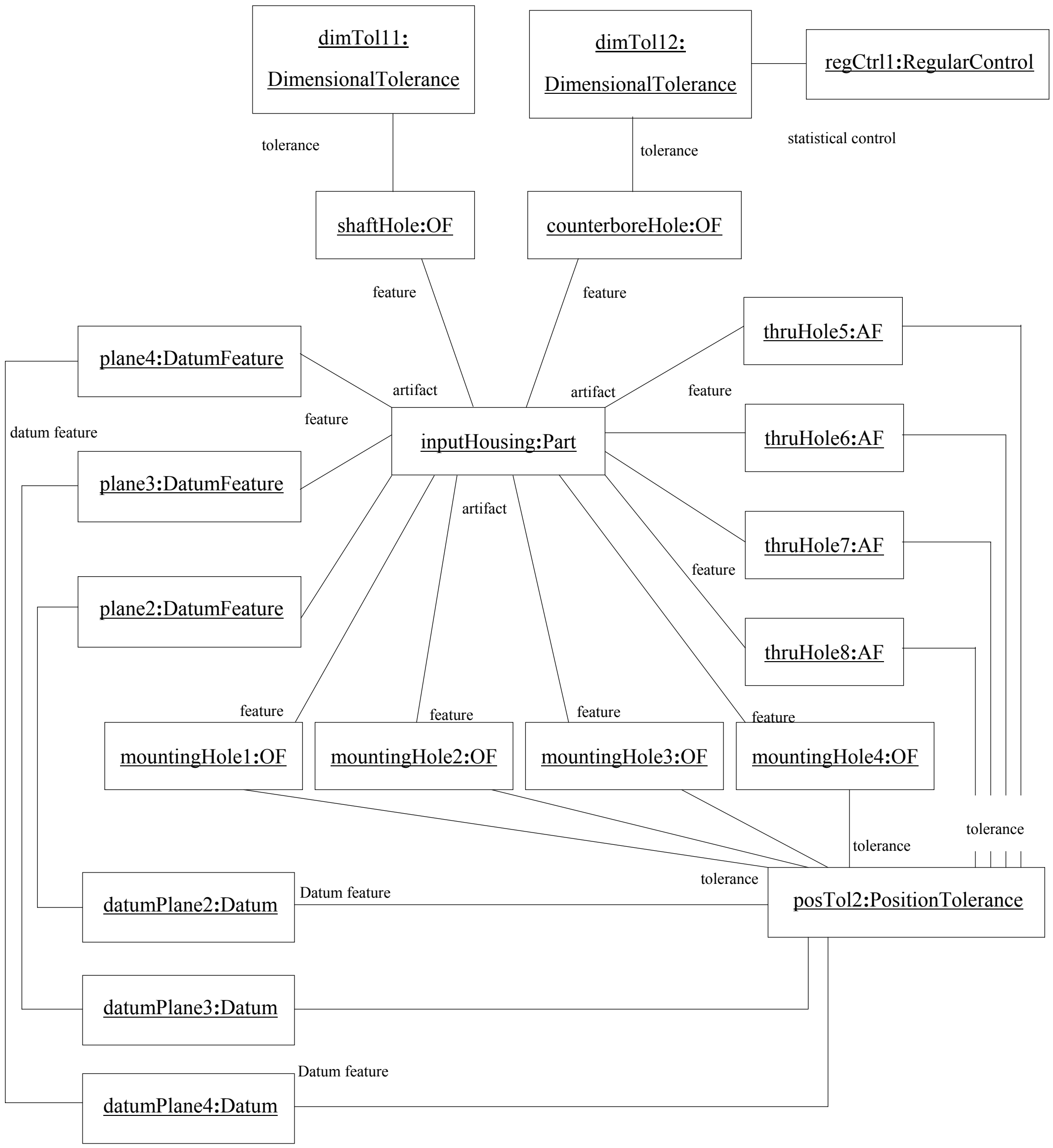


Figure 67: Input Housing Part Instance Diagram

Figure 68 shows the engineering drawing of the output housing. Figure 69 has the tolerance instance diagram, which shows how the tolerances are represented using the UML model discussed in Section 3.11. Figure 70 shows an integrated view of the part, its tolerances, and the assembly features. Table 19 tabulates the features, tolerances, and control frames with the artifact association.

In the cross-sectional view in Figure 68, the hole with a diameter of the 28.58 mm has a statistical control on the dimension and another statistical control in the position tolerance. Because this hole will contain bearings, both dimension and position have to be toleranced to ensure a tight fit. The depth of the hole is not controlled since it is not critical to the fit between this part and bearings. Datum **A** to be established from the bottom surface is shown in the front view. Datums **B** and C are shown in the bottom view of the input housing. Also, the direction and location of the cross-section is shown in this view. One position tolerance is applied to a pattern of four holes, each with the diameter of 5.38 mm. The position is measured in a reference to a coordinate system established by Datums A, B, and C. The tolerance is at the maximum material condition of the hole. Another position tolerance is applied to another pattern of four 7.19 mm holes. The tolerance is also at the maximum condition of the hole. This tolerance is statistically controlled because the positions of these holes are critical in assembly. The output housing and the ring gear have to be properly aligned and assembled. For alignment, there are two pin holes to assist the alignment. Their position is also similarly toleranced with an instance of **StatisticalControl.**

Features	**Tolerances**	**Feature Control Frames**	**Artifact Association**
Bottom plane of mounting plate (Datum A) (plane2:DatumFeature)	None	None	None
Side plane of mounting plate (Datum B) (plane3:DatumFeature)	None	None	None
Side plane of mounting plate (Datum C) (plane4:DatumFeature)	None	None	None
Four through holes* (thruHole5~8:AF)	Position tolerance	⌖ | 0.05 | A | B | C (posTol2:PositionTolerance)	*fc13*
Four mounting holes (mountingHole1~4:OF)	Position tolerance	⌖ | 0.05 | A | B | C (posTol2:PositionTolerance)	None
Shaft hole (ShaftHole:OF)	Dimensional tolerance	(ϕ25.53) ± 0.01 (dimTol11:DimensionalTolerance)	None
Counterbore hole (CounterboreHole:OF)	Dimensional tolerance	ϕ37.64 ~ ϕ38.23 ⟨ST⟩ (dimTol12:DimensionalTolerance)	None

Table 18: Tolerance Table For The Input Housing

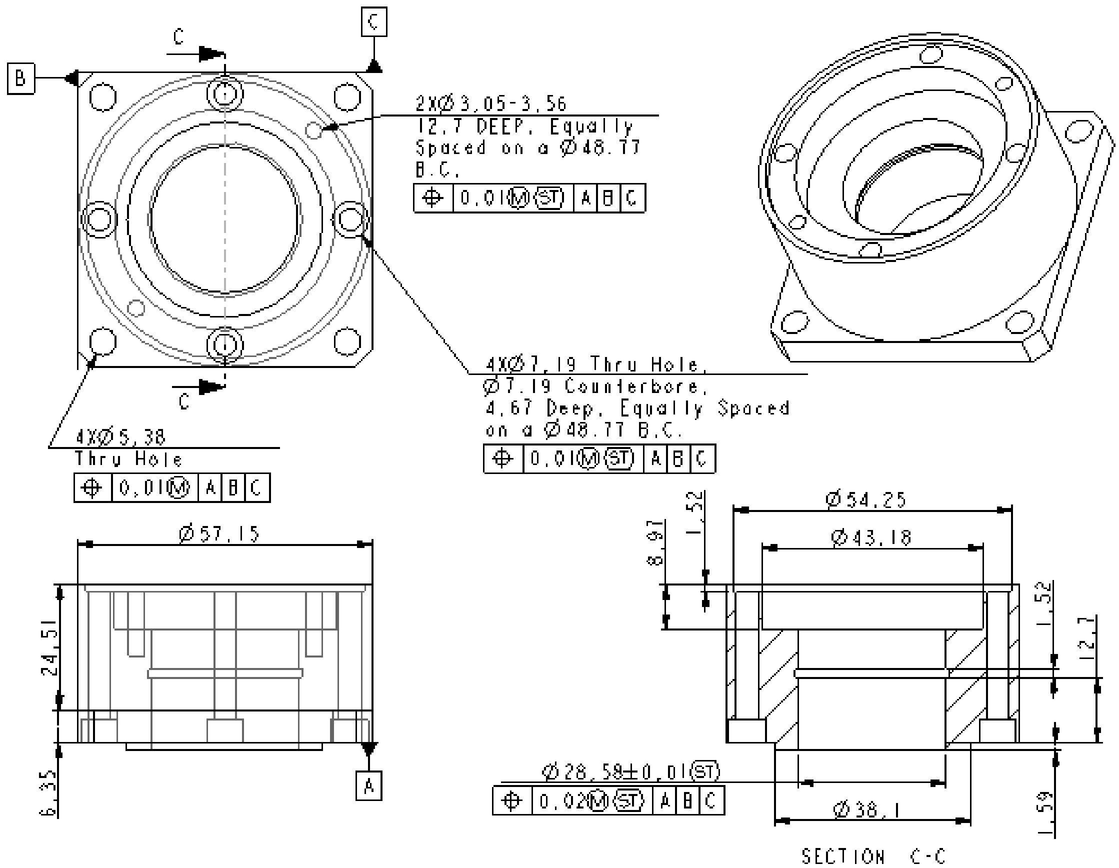


Figure 68: Output Housing Tolerances

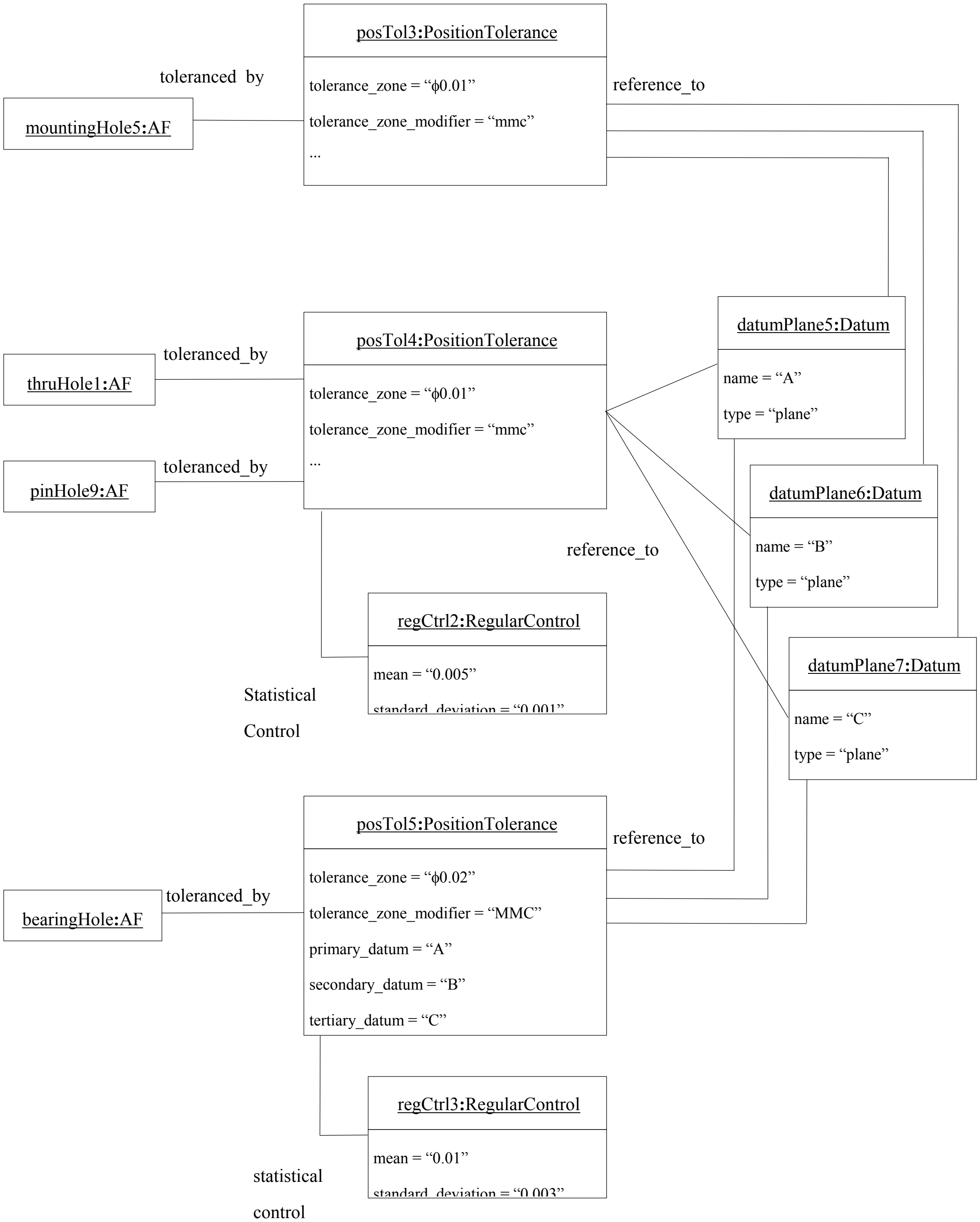


Figure 69: Tolerance Instance Diagram

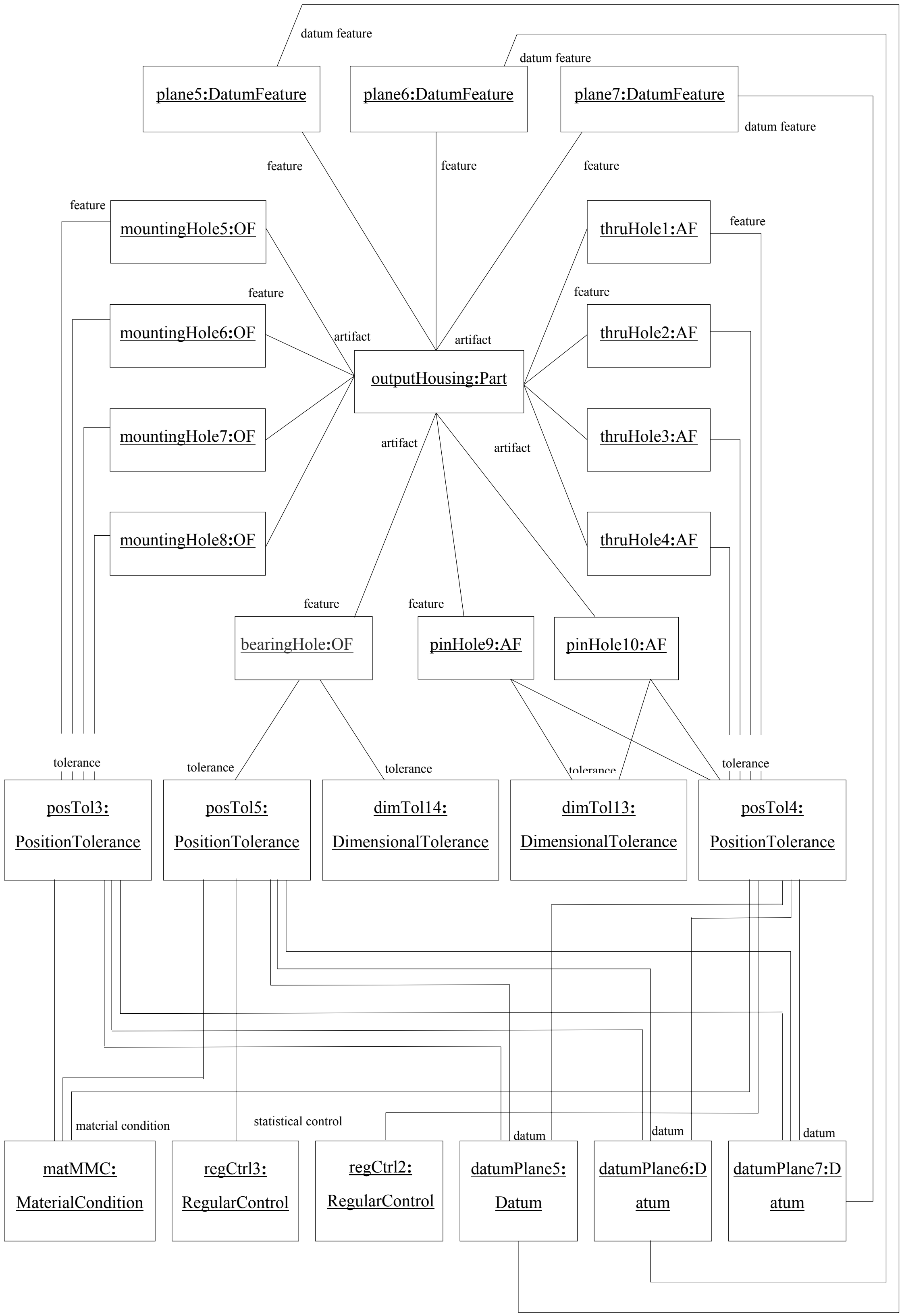


Figure 70: Output Housing Instance Diagram

Both dimensional and geometric tolerances are described in these two subsections. The tolerances can be captured by objects instanced from the appropriate tolerance classes, specified in Section 3.11.

Features	**Tolerances**	**Feature Control Frames**	**Artifact Association**
Bottom plane of mounting plate (Datum A) (plane5:DatumFeature)	None	None	None
Side plane of mounting plate (Datum B) (Plane6:DatumFeature)	None	None	None
Side plane of mounting plate (Datum C) (Plane7:DatumFeature)	None	None	None
Four mounting holes (mountingHole5~8:OF)	Position tolerance	⌖ 0.01 Ⓜ A B C (posTol3:PositionTolerance)	None
Four through holes* (thruHole1~4:AF)	Position tolerance	⌖ 0.01 Ⓜ ⟨ST⟩ A B C (posTol4:PositionTolerance)	*fc11*
Two pin-hole surfaces* (pinHole9~10:AF)	Position tolerance	⌖ 0.01 Ⓜ ⟨ST⟩ A B C (posTol4:PositionTolerance)	*fc10*
	Dimensional tolerance	ϕ3.05 ~ ϕ3.56 (dimTol13:DimensionalTolerance)	
Bearing hole (bearingHole:OF)	Position tolerance	⌖ 0.02 Ⓜ ⟨ST⟩ A B C (posTol5:PositionTolerance)	None
	Dimensional tolerance	(ϕ28.58) ± 0.01 ⟨ST⟩ (dimTol14:DimensionalTolerance)	

Table 19: Tolerance Table For The Output Housing

5 Conclusions and Future Work

In this report, 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 [4]. 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 [12]

6 Acknowledgements

The authors wish to acknowledge the valuable comments and improvements suggested by Prof. Steven Fenves. Those comments have substantially improved and shaped the report. 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

7 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 report 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.

8 References

1. ISO TC 194/SC4, "*Official TC184/SC4 Web Site*," http://www.tc184-sc4.org/, 2003.

2. Nobuhiro Sugimura, and Akihiko Ohtaka, "ISO TC 184/SC4/WG12 N597, JNC Proposal of STEP Assembly Model for Products (June 2000).," ISO, 2000.

3. R.Sudarsan, Y.Narahari, U.Roy, R.D.Sriram, K.W.Lyons, and N.Pramanik, "*Information Models for Design Tolerancing: from conceptual to the detail design*," National Institute of Standards and Technology, Gaithersburg, MD 20899, USA, 1999.

4. Steven J.Fenves, " *A Core Product Model For Representing Design Information*," National Institute of Standards and Technology, NISTIR6736 , Gaithersburg, MD 20899, USA, 2001.

5. K.W.Lyons, V.N.Rajan, and R.Sreerangam, "*Representations and Methodologies for Assembly Modeling*.," National Institute of Standards and Technology, NISTIR 6059, Gaithersburg, MD 20899, USA, 1996.

6. Venkat N.Rajan, Kevin W.Lyons, and Raj Sreerangam, " *Assembly Representations for Capturing Mating Constraints and Component Kinematics*," Proceedings of the 1997, IEEE International Symposium on Assembly and Task Planning, Marina del Rey, CA (August 1997)., 1997.

7. K.Lyons, S.Shooter, W.Keirouz, and P.Hart, "*The Open Assembly design environment: an architecture for design agent interoperability*," ASME, 1998.

8. G.Booch, J.Rumbaugh, and I.Jacobson, *The United Modeling Language User Guide*, Addison-Wesley 1997.

9. Kemmerer, S., "STEP: The Grand Experience, (Editor)," NIST Special Publication 939, National Institute of Standards and Technology, Gaithersburg, MD 20899, USA, 1999.

10. Philip R.Kennicott, "*ISO TC 184/SC4: Product Data Representation and Exchange, Part: 44, Title: Industrial Automation Systems and Integration Product Data Representation and Exchange – Integrated Generic Resources: Product Structure Configuration* (November 1994).," ISO , 1994.

11. Ram D Sriram, "*Standards for the Collaborative Design Enterprise, Response to Object Management Group's (OMG) Mfg DTG RFI #4*," 1999.

12. Connacher, H., Sankar Jayaram, and Kevin W Lyons, "*Virtual assembly using virtual reality techniques* ," *Computer-Aided Design*, Vol. 29, No. 8, 1997, pp. 575-584.

13. U.Roy, R.Sudarsan, R.D.Sriram, K.W.Lyons, and M.R.Duffey, "*Information Architecture for Design Tolerancing: from Conceptual to the Detail Design*," Proc. of DETC'99, I999 ASME International Design Engineering Technical Conferences, Nevada, Las Vegas, USA , 1999.

14. MOKA. *MOKA: A Framework for structuring and representing engineering knowledge* . http://www.kbe.coventry.ac.uk/moka/miginfo.htm . 1999.

15. Whitney, D. E., and Mantripragada, R., "*The Datum Flow Chain: A systematic approach to assembly design and modeling* .," Proceedings of the 1998 ASME Design Engineering Technical Conferences and Computers in Engineering Conference', ASME (1998), 1998.

16. Heissermann, J., and Mattikalli, R., "*Representing relationships in hierarchical assemblies*," Proceedings of the 1998 ASME De-sign Engineering Technical Conferences and Computers in Engineering Conference', ASME (1998, 1998.

17. Van der Net, A.. *Designing and manufacturing assemblies*. 1998. Eindhoven University of Technology .

18. P.N.Sheth and J.J.Uicker, " *A Generalized Symbolic Notation for Mechanisms*," *Trans.ASME Journal of Engineering for Industry*, Vol. 93, No. 1, 1971, pp. 102-112.

19. ASME, *Dimensioning and Tolerancing Y14.5M* , ASME 1994.

Appendix A SU Parameters And Linear Transformation Matrix

The SU-parameters (Sheth-Uicker parameters; see reference [18]) mean six parameters to describe the spatial relation (position and orientation) between two coordinate systems (frames). By describing the spatial relation between the two frames, the shape of a kinematic link needed for kinematic analysis can be defined. In fact, the exact geometry of the link is not required.

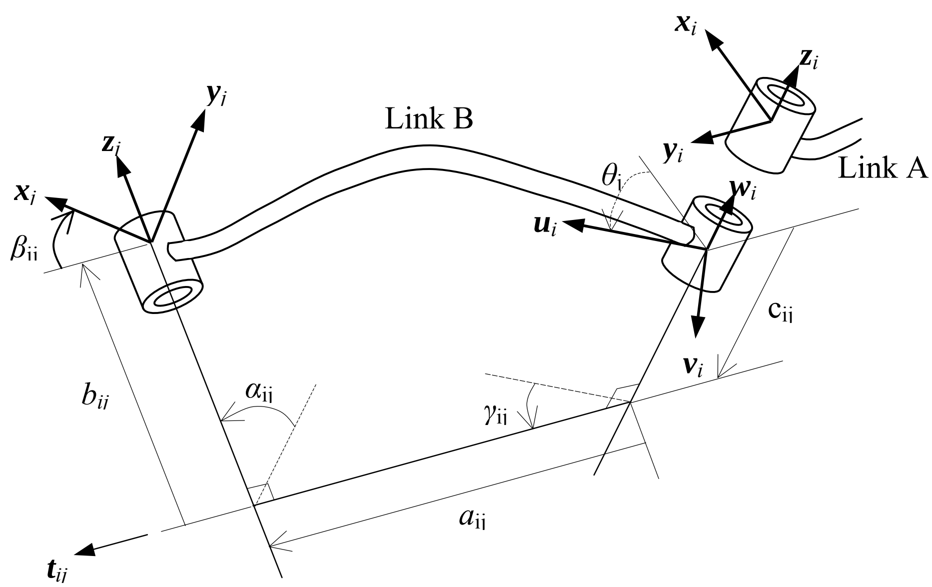


Figure A-1: Definition of SU-Parameters

Figure A-1 shows the SU-parameters defined for Link B. The frame $\{u_i v_i w_i\}$ is established at the beginning end of the link in a kinematic loop, and frame $\{x_j y_j z_j\}$ at the following end. To define the parameters, the common perpendicular is found first between two axes w_i and z_j. The vector $\boldsymbol{t}_{ij}$ in Figure A-1 shows the common perpendicular found. Then the six parameters are defined with the following conventions.

a_{ij} = distance from $\boldsymbol{w}_i$ to $\boldsymbol{z}_j$ along $\boldsymbol{t}_{ij.}$

α_{ij} = angle from $\boldsymbol{w}_i$ to $\boldsymbol{z}_j$ along the counterclockwise direction about $\boldsymbol{t}_{i,}$

b_{ij} = distance from the common perpendicular to the origin of $\{x_j y_j z_j\}$ along $\boldsymbol{z}_{j,}$

β_{ij} = angle from $\boldsymbol{t}_{ij}$ to $\boldsymbol{x}_j$ along the counterclockwise direction about $\boldsymbol{z}_j$,
c_{ij} = distance from the origin of $\{u_i v_i w_i\}$ to the common perpendicular along $\boldsymbol{w}_i$,
γ_{ij} = angle from $\boldsymbol{u}_i$ to $\boldsymbol{t}_{ij}$ along the counterclockwise direction about $\boldsymbol{w}_i$.

Once the SU-parameters are defined between two frames, we can derive the 4 × 4 linear transformation matrix as follows:

$$\mathbf{T}_{ij} = \begin{bmatrix} \cos\gamma_{ij}\cos\beta_{ij} - \sin\gamma_{ij}\cos\alpha_{ij}\sin\beta_{ij} & -\cos\gamma_{ij}\sin\beta_{ij} - \sin\gamma_{ij}\cos\alpha_{ij}\cos\beta_{ij} & \sin\gamma_{ij}\sin\alpha_{ij} & a_{ij}\cos\gamma_{ij} + b_{ij}\sin\gamma_{ij}\sin\alpha_{ij} \\ \sin\gamma_{ij}\cos\beta_{ij} + \cos\gamma_{ij}\cos\alpha_{ij}\sin\beta_{ij} & -\sin\gamma_{ij}\sin\beta_{ij} + \cos\gamma_{ij}\cos\alpha_{ij}\cos\beta_{ij} & -\cos\gamma_{ij}\sin\alpha_{ij} & a_{ij}\sin\gamma_{ij} - b_{ij}\cos\gamma_{ij}\sin\alpha_{ij} \\ \sin\alpha_{ij}\sin\beta_{ij} & \sin\alpha_{ij}\cos\beta_{ij} & \cos\alpha_{ij} & c_{ij} + b_{ij}\cos\alpha_{ij} \\ 0 & 0 & 0 & 1 \end{bmatrix}.$$

The SU parameters for a kinematic pair can also be defined in a similar manner. The difference is that there is one variable parameter among the six parameters. This variable is called *pair variable*. Due to the pair variable, the final linear transformation matrix will also have variable elements. In Figure A-1, the link A and B forms a revolute pair. The SU-parameters between frame $\{x_i y_i z_i\}$ and $\{u_i v_i w_i\}$ can be derived using the same convention mentioned previously. One of the parameters γ_{ij} and β_{ij} may be the pair variable θ_j, as shown in Figure A-1.

Appendix B Derivation Of Kinematic Equations Of Planetary Gear System

To derive the kinematic equation of the planetary gear system in Figure 46, we should first obtain the matrix equation of each loop. To this end, the constant shape matrix and variable pair matrix should first be obtained.

The constant shape matrix refers to the linear transformation matrix between two coordinate systems (frames) of the rigid link. To derive the transformation matrix, six SU-parameters are first determined (they are constants for a rigid link). Once the SU-parameters are obtained, the linear transformation matrix can be derived straightforward. The variable pair matrix describes the linear transformation of a kinematic pair: i.e., the motion of the joint. It can be derived in the same way as the constant shape matrix, except that it contains variable elements originating from its pair variable. Details of the notion of SU-parameters and derivation of the corresponding transformation matrix for each type of kinematic pairs can be found in Reference [18].

With reference to Figure 46(a) and Table 12 the first kinematic loop and associated frames can be derived as follows:

$$\begin{array}{ccccccc} \text{Support} & & \text{Sun Gear } (\mathbf{T}_{12}) & & \text{Planet Gear } (\mathbf{T}_{23}) & & \text{Ring Gear } (\mathbf{T}_{31}) \\ \{x_1 y_1 z_1\} & \Rightarrow & \left[\{u_1 v_1 w_1\} \rightarrow \{x_2 y_2 z_2\}\right] & \Rightarrow & \left[\{u_2 v_2 w_2\} \rightarrow \{x_3 y_3 z_3\}\right] & \Rightarrow & \left[\{u_3 v_3 w_3\} \rightarrow \{x_1 y_1 z_1\}\right], \\ & \mathbf{\Phi}_1(\theta_1) & & \mathbf{\Phi}_2(\zeta_2, \theta_2) & & \mathbf{\Phi}_3(\zeta_3, \theta_3) & \end{array}$$

where $\mathbf{\Phi}_i$ is a variable pair matrix, $\mathbf{T}_{ij}$ is a const shape matrix, θ_i is a pair variable, and ζ_i is a gear ratio. The SU-parameters and corresponding transformation matrices are shown in Table B-1.

Link or Pair	**SU-parameters and Transformation Matrix**
Revolute Pair 1	a = 0, α = 0, b = 0, β = 0, c = 0, γ = θ_1 $\mathbf{\Phi}_1(\theta_1) = \begin{bmatrix} \cos\theta_1 & -\sin\theta_1 & 0 & 0 \\ \sin\theta_1 & \cos\theta_1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$
Sungear	a = 0, α = 0, b = 0, β = 0, c = 0, γ = 0

Link or Pair	SU-parameters and Transformation Matrix
	$\mathbf{T}_{12} = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$
Gear Pair 1	$a = R_1+R_2$, $\alpha = 0$, $b = 0$, $\beta = \zeta_2\theta_2$, $c = 0$, $\gamma = \theta_2$ $\mathbf{\Phi}_2(\zeta_2,\theta_2) = \begin{bmatrix} \cos(1+\zeta_2)\theta_2 & -\sin(1+\zeta_2)\theta_2 & 0 & (1+\zeta_2)R_2\cos\theta_2 \\ \sin(1+\zeta_2)\theta_2 & \cos(1+\zeta_2)\theta_2 & 0 & (1+\zeta_2)R_2\sin\theta_2 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}, \quad \zeta_2 = \frac{R_1}{R_2}$
Planetary Gear	$a = 0, \alpha = 0, b = 0, \beta = 0, c = 0, \gamma = 180°$ $\mathbf{T}_{23} = \begin{bmatrix} -1 & 0 & 0 & 0 \\ 0 & -1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$
Gear Pair 4	$a = R_3-R_2, \alpha = 0, b = 0, \beta = \zeta_3\theta_3, c = 0, \gamma = \theta_3$ $\mathbf{\Phi}_3(\zeta_3,\theta_3) = \begin{bmatrix} \cos(1+\zeta_3)\theta_3 & -\sin(1+\zeta_3)\theta_3 & 0 & (1+\zeta_3)R_3\cos\theta_3 \\ \sin(1+\zeta_3)\theta_3 & \cos(1+\zeta_3)\theta_3 & 0 & (1+\zeta_3)R_3\sin\theta_3 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}, \quad \zeta_3 = -\frac{R_2}{R_3}$
Ring Gear	$a = 0, \alpha = 0, b = 0, \beta = 0, c = 0, \gamma = 0$ $\mathbf{T}_{31} = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$

Table B-1: SU-Parameters And Transformation Matrices Of The First Loop

Considering that the first loop closes on itself (it starts from the ground of a unknown support and ends in the ground of fixed ring gear), the following matrix equation can be found:

$$\mathbf{\Phi}_1(\theta_1)\mathbf{T}_{12}\mathbf{\Phi}_2(\zeta_2,\theta_2)\mathbf{T}_{23}\mathbf{\Phi}_3(\zeta_3,\theta_3)\mathbf{T}_{31} = \mathbf{I}, \tag{B-1}$$

where **I** is an identity matrix. Using Eq. (B-1) and the shape and pair matrices in Table B-1, the following equation is obtained:

$$\begin{bmatrix} -\cos P & \sin P & 0 & -(1+\zeta_3)R_3\cos Q + (1+\zeta_2)R_2\cos R \\ \sin P & -\cos P & 0 & -(1+\zeta_3)R_3\sin Q + (1+\zeta_2)R_2\sin R \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} = \mathbf{I}, \tag{B-2}$$

where $P=\theta_1+(1+\zeta_2)\theta_2+(1+\zeta_3)\theta_3$, $Q=\theta_1+(1+\zeta_2)\theta_2+\theta_3$, *and* $R=\theta_1+\theta_2$.

The kinematic equation for the second loop can also be derived in a similar manner. With reference to Figure 46 (b), the second kinematic loop and associated frames can be derived as follows:

$$\begin{array}{ccccccc} \text{Bearing} & & \text{Planet Carrier } (\mathbf{T}_{45}) & & \text{Planet Gear } (\mathbf{T}_{53}) & & \text{Ring Gear } (\mathbf{T}_{34}) \\ \{x_4y_4z_4\} & \Rightarrow & [\{u_4v_4w_4\}\rightarrow\{x_5y_5z_5\}] & \Rightarrow & [\{u_5v_5w_5\}\rightarrow\{x_3y_3z_3\}] & \Rightarrow & [\{u_3v_3w_3\}\rightarrow\{x_4y_4z_4\}], \\ & \mathbf{\Phi}_4(\theta_4) & & \mathbf{\Phi}_5(\theta_5) & & \mathbf{\Phi}_3(\zeta_3,\theta_3) & \end{array}$$

where $\mathbf{\Phi}_i$ is a variable pair matrix, $\mathbf{T}_{ij}$ is a constant shape matrix, θ_i is a pair variable, and ζ_i is a gear ratio. The SU-parameters and corresponding transformation matrices are shown in Table B-2.

Link or Pair	**SU-parameters and Transformation Matrix**
Revolute Pair 5	a = 0, α = 0, b = 0, β = 0, c = 0, γ = θ_4 $\mathbf{\Phi}_4(\theta_4)=\begin{bmatrix}\cos\theta_4 & -\sin\theta_4 & 0 & 0\\ \sin\theta_4 & \cos\theta_4 & 0 & 0\\ 0 & 0 & 1 & 0\\ 0 & 0 & 0 & 1\end{bmatrix}$
Planet Carrier	a = L, α = 0, b = 0, β = 0, c = 0, γ = 0 $\mathbf{T}_{45}=\begin{bmatrix}1 & 0 & 0 & L\\ 0 & 1 & 0 & 0\\ 0 & 0 & 1 & 0\\ 0 & 0 & 0 & 1\end{bmatrix}$, $L = R_1 + R_2$
Revolute Pair 2	a = 0, α = 0, b = 0, β = 0, c = 0, γ = θ_5 $\mathbf{\Phi}_5(\theta_5)=\begin{bmatrix}\cos\theta_5 & -\sin\theta_5 & 0 & 0\\ \sin\theta_5 & \cos\theta_5 & 0 & 0\\ 0 & 0 & 1 & 0\\ 0 & 0 & 0 & 1\end{bmatrix}$
Planetary Gear	a = 0, α = 180°, b = 0, β = 0, c = 0, γ = 180° $\mathbf{T}_{23}=\begin{bmatrix}-1 & 0 & 0 & 0\\ 0 & 1 & 0 & 0\\ 0 & 0 & -1 & 0\\ 0 & 0 & 0 & 1\end{bmatrix}$
Gear Pair 4	a = R_3-R_2, α = 0, b = 0, β = $\zeta_3\theta_3$, c = 0, γ = θ_3 $\mathbf{\Phi}_3(\zeta_3,\theta_3)=\begin{bmatrix}\cos(1+\zeta_3)\theta_3 & -\sin(1+\zeta_3)\theta_3 & 0 & (1+\zeta_3)R_3\cos\theta_3\\ \sin(1+\zeta_3)\theta_3 & \cos(1+\zeta_3)\theta_3 & 0 & (1+\zeta_3)R_3\sin\theta_3\\ 0 & 0 & 1 & 0\\ 0 & 0 & 0 & 1\end{bmatrix}$, $\zeta_3=-\dfrac{R_2}{R_3}$
Ring Gear	a = 0, α = 180°, b = 0, β = 0, c = 0, γ = 180°

Link or Pair	SU-parameters and Transformation Matrix
	$\mathbf{T}_{31}=\begin{bmatrix}-1 & 0 & 0 & 0\\ 0 & 1 & 0 & 0\\ 0 & 0 & -1 & 0\\ 0 & 0 & 0 & 1\end{bmatrix}$

Table B-2: SU-Parameters And Transformation Matrices Of The Second Loop

Considering the second loop closes itself, the following matrix equation can be found:

$$\mathbf{\Phi}_4(\theta_4)\mathbf{T}_{45}\mathbf{\Phi}_5(\theta_5)\mathbf{T}_{53}\mathbf{\Phi}_3(\zeta_3,\theta_3)\mathbf{T}_{34}=\mathbf{I}, \tag{B-3}$$

where **I** is an identity matrix. Using Eq. (B-3) and the shape and pair matrices in Table B-2, the following equation is obtained:

$$\begin{bmatrix}\cos S & -\sin S & 0 & -(1+\zeta_3)R_3\cos T+L\cos\theta_4\\ \sin S & \cos S & 0 & -(1+\zeta_3)R_3\sin T+L\sin\theta_4\\ 0 & 0 & 1 & 0\\ 0 & 0 & 0 & 1\end{bmatrix}=\mathbf{I}, \tag{B-4}$$

where $S=\theta_4+\theta_5-(1+\zeta_3)\theta_3$, $T=\theta_4+\theta_5-\theta_3$, and $L=R_1+R_2$.

The previously derived equations. (B-2) and (B-4) are the kinematic equations of the planetary gear system. By solving equations (B-2) and (B-4), we can obtain the angular displacement relations of the gear system. Equating the elements of the left-hand sides of equations (B-2) and (B-4) to the corresponding elements of the right-hand sides of identity matrix, the following relations are obtained (θ_1 is considered here as an input and thus being known):

$$\begin{aligned}\theta_2&=-\frac{1}{\zeta_2}\theta_3,\\ \theta_3&=\frac{\zeta_2}{1-\zeta_2\zeta_3}(\theta_1-\pi),\\ \theta_4&=\frac{\zeta_2\zeta_3}{1-\zeta_2\zeta_3}(\theta_1-\pi),\\ \theta_5&=\theta_3.\end{aligned} \tag{B-5}$$

NOTES