5-2012

Functional and Behavioral Product Information Representation and Consistency Validation for Collaboration in Product Lifecycle Activities

Mehmet Murat Baysal
Syracuse University

Abstract

Information models that represent the function, assembly and behavior of artifacts are critical in the conceptual development of a product and its evaluation. Much research has been conducted in this area; however, existing models do not relate function, behavior and structure in a comprehensive and consistent way. In this work, NIST's Core Product Model (CPM) and the Open Assembly Model (OAM) are extended to integrate product information including function and behavior, with an emphasis on assembly, throughout all phases of product development. For function and flow classification, the NIST functional taxonomy is used to maintain consistency with the literature.

The consistency validation of product information, and the verification of modified product information are discussed; these processes ensure that the product information has no contradictions and allows tracing through associations without any deficiency or disconnection. In other words, the information model has to be complete in terms of traceability of function, behavior, spatial relationships, etc., in order to support all information exchange activities. The product information representation provides a mechanism for capturing product information and storing it in a database. This representation schema also provides necessary information for any future decision making activities in the End of Life (EOL) environment, such as the replacement or reuse of any part or subassembly. When there is a need to replace one artifact with another, one must consider all of the associations of the existing artifact with other artifacts and the environment, not just functional and space requirements, and the relevant modification(s) of the associated objects has to verified. So one can manage product lifecycle activities in different perspectives by knowing how the product information is interconnected in various domains and how its characteristics affect each other.

FUNCTIONAL AND BEHAVIORAL PRODUCT INFORMATION REPRESENTATION AND CONSISTENCY VALIDATION FOR COLLABORATION IN PRODUCT LIFECYCLE ACTIVITIES

By

Mehmet Murat Baysal

B.S. Cukurova University, Turkey
M.S. Cukurova University, Turkey

DISSERTATION

Submitted in partial fulfillment of the requirements for the degree
of Doctor of Philosophy in Mechanical and Aerospace Engineering
in the Graduate School of Syracuse University

May 2012

Acknowledgements

I wish to express my sincere gratitude to my academic advisor, Professor U. Roy, for his support and encouragement throughout my program of study and research work. It has been a pleasure to work with him.

I want also to acknowledge my gratitude to Dr. Riyad S. Aboutaha for chairing my oral defense committee. And I would like to thank Dr. Achille Messac, Dr. John F. Dannenhoffer, Dr. Eric M. Lui, and Dr. Rachuri Sudarsan for consenting to serve as members of my dissertation committee. I also thank Dr. Ram D Sriram of the National Institute of Standards and Technology for funding my research at the beginning.

Further thanks go to Mehmet Ilteris Sarigecili for his constant support, for fundamental and theoretical discussions, and for his companionship and encouragement; to my friends Ahmet Sula, Sevki Unal and Lowel Lingo for their support and many meaningful discussions; to Muris Hadzic for proofreading; to Dr. Ehat Ercanli for his valuable suggestions and support; to Cukurova University; and especially to Dr. Huseyin Akilli and Dr. Besir Sahin for their faith in me and for supporting me in coming to Syracuse University.

Finally, and most important, I would like to thank my wife Seher and my daughters Ayse Sila and Ruveyda Gul for their infinite love and patience, without which it would have been impossible for me to complete my research work. I thank my parents, Ayse and Kasim, my brother Ali, and my sister Nuray for their faith in me, and their constant supports and prayers which gave me ability to tackle challenges head on. Also, I would like to thank Seher's parents, Hatice and Ramazan Kayacik for their constant support.

Table of Contents

List of Figures

List of Tables

Chapter 1

Introduction

1.1 Introduction and Motivation

In today's marketplace, the realities of globalization and competition, as well as the complexity of modern products, are forcing companies to distribute product life cycle activities (i.e., product development, manufacturing and assembly, etc.) across different stakeholders located globally. In addition, the stakeholders exchange not only geometric data, but also knowledge about design and product processes, the functions and behaviors of the product, and design intent. As product development becomes increasingly knowledge-intensive and collaborative, support for the representation and exchange of product information becomes more important for collaboration in the product life cycle activities. For example, a study by the NIST Strategic Planning and Economic Assessment Office conservatively estimated the economic losses due to lack of interoperability in the US automotive supply chain alone at $1.05 billion per year (Brunnermeier, 1999). In a broader aspect, NIST Advanced Technology Program published a report about "Cost Analysis of Inadequate Interoperability in the U.S. Capital Facilities Industry" (Gallaher, 2004). This report conservatively estimates the annual interoperability costs as $15.8 billion in the U.S. capital facilities industry, which is the 1-2 % of industry revenue, in 2002.

In product representation, the extraction of geometry information from solid models has always been straightforward; however, it is necessary to identify semantic structures (i.e., features) for solid reasoning about a component's function. There are a number of methods and techniques for delineating a functional structure, and some efforts have been made to connect function and behavior with structure. However, we cannot say that there is complete, correct and consistent product information representation. Therefore, it is necessary to develop a means of representation for function, behavior and structure data models that will provide more correct and consistent product information to all agents in the product lifecycle. Representing assembly information, including information about product functions and behavior, and providing a mechanism for exchanging product information throughout the lifecycle of a product will facilitate efficient collaboration among different stakeholders and reduce interoperability costs and product development time.

Figure 1.1 shows a variety of tools used by stakeholders (e.g., designers, analysts, etc.) in different phases of product lifecycle activities. It also shows how product information flows through product life cycle activities (shown with blue arrows on the left), starting from transforming customer needs, moving to product requirements and engineering specifications, and finally to the disposal of the product in conventional way. Then, based on these requirements, functions are defined in the functional design stage.

The next step is to generate appropriate concepts (the conceptual design stage) that determine the overall product geometry, the material properties for individual parts and the assembly, and any kinematic synthesis. After evaluation of the appropriate concepts, one concept is selected among alternatives.

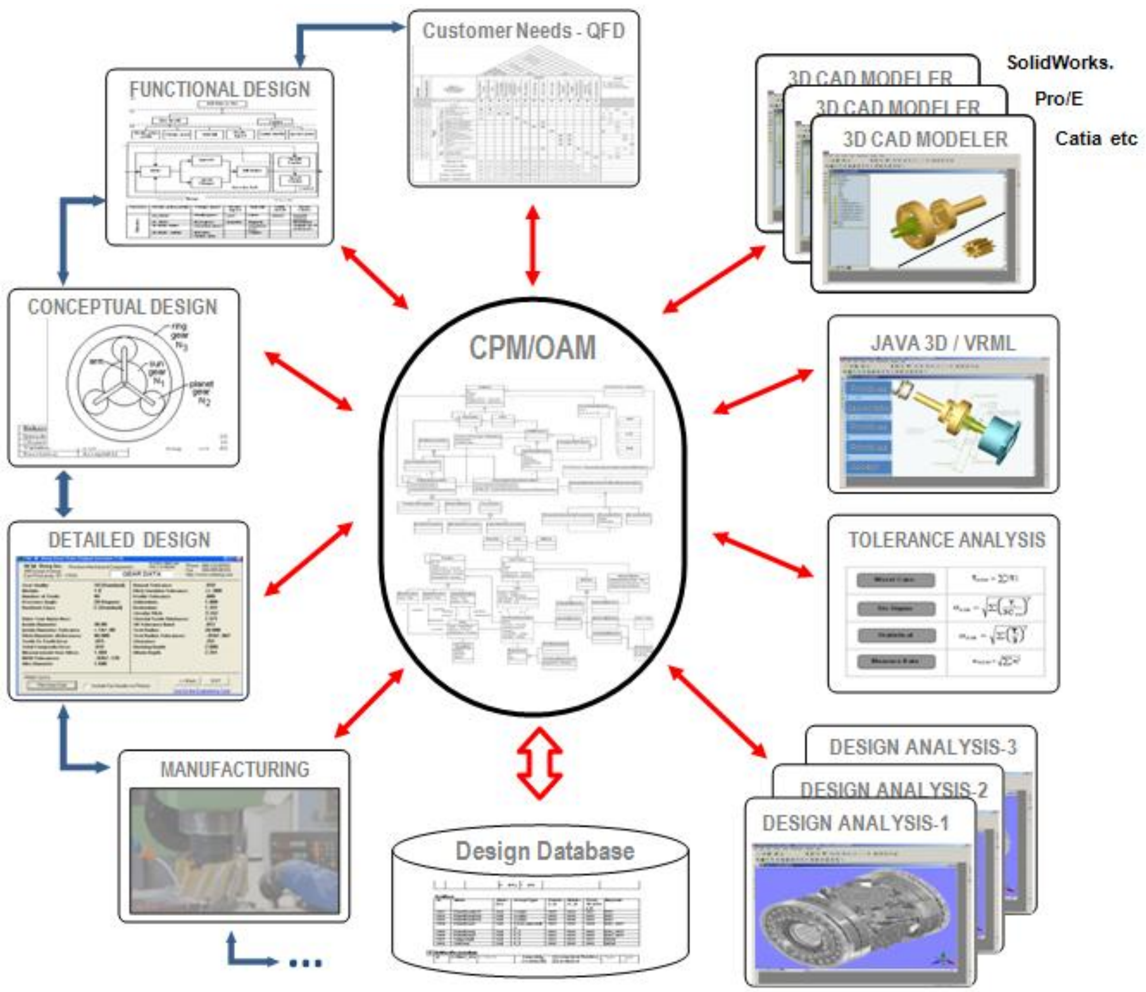


Figure 1.1: The Use of the Product Information Model in Product Lifecycle Activities

Then the detailed design (for detailed geometry, materials, tolerances, etc.) is outlined, followed by analysis, manufacturing, assembly, inspection, and so forth. In conventional product development, every stage inherits information from the previous stage and provides information to the next stage. The results of each stage are checked according to the inputs from the previous stage. This means that the product information flows in two directions. Sometimes, information flows between other stages as well. The information flow in this study is shown with the red arrows (in the center of Figure 1.1), and, as can be seen, for each stage information is drawn from the product information model and delivered to it, along with interrelationships among objects in the product information model. In order to exchange information, it must be standardized.

Figure 1.2 shows various product information classes, including already standardized, partially standardized, and non-standardized product information, and process and analysis information. Geometry and topology have already been standardized by different institutions and made available to the public. Geometric and dimensional tolerances and assembly relationships are also represented in standardized ways, but there are no information exchange translators for other product lifecycle activity tools. Although standalone standards and applications are available for some process and analysis activities, means of connecting with other lifecycle activities are lacking. Many efforts have been made by researchers to represent product requirements, functions, behavior, and design intent, but there is no standard representation and there are no well-defined connections with other product information, such as product structure.

This study primarily focuses on the standard representation of product information—mainly product structure, assembly relationships, kinematics, tolerance, function, and behavior, and the interrelations among them. Design intent and requirements are just introduced into the model, but left for the future study.

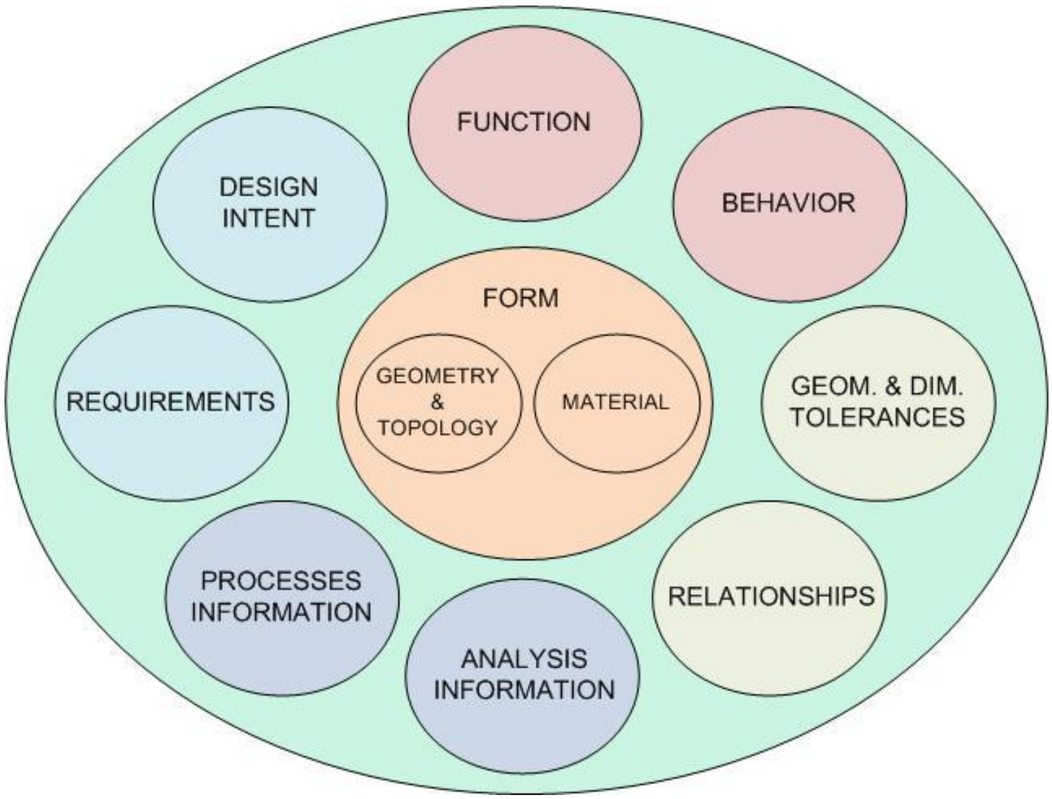


Figure 1.2: Product Information Domains in a Product Information Model

There are many associations between any two parts of a product, in terms of assembly, function, behavior, tolerance, kinematics, etc. These associations need to be represented in a consistent way, so that they will not conflict with each other. If position, orientation, joint type, and so forth, in the assembly are changed for any reason, the function and behavior will be affected accordingly. In that case, the function and behavior model can be modified to consider the new information, since all inputs and outputs are defined in terms of the associations.

Although there is much research on functional and behavioral representation, there are no common definitions for function or behavior. In this work, first the function and behavior information of a product are defined, then the associations among functions, behaviors and artifacts in the assembly. In this functional model, the function is initially defined in terms of the overall function. Then, the overall function is decomposed into, and supported by, sub-functions. Each function has a certain priority. In this way, we ensure that a certain function has to wait until the prior one(s) is processed. Another property of function is functional associations, which define the relationships among the artifacts and the behavior of the artifacts, based on these associations. The more important feature of this model is that function, behavior and assembly information are interrelated. Function and behavior are related through functional associations, and behavior and artifact are related through a behavioral model. Therefore, if any change is required or a problem occurs, the designer will be able to (1) check the intention behind that feature, artifact or any artifact association in the assembly, (2) see how it will affect the other entities, and (3) trace any problem through the associations. In order to accomplish them, first, it is necessary to represent product information and interrelationships in product lifecycle activities. Second one is to develop a product information model that provides information exchange without compromising consistency of the product information.

1.1.1 Product Information Representation

The first goal to enable efficient collaboration among different stakeholders through product life cycle activities is to develop product information representation for the assembly structure, product function, behavior and design intent. For this reason, Core Product Model (CPM) and Open Assembly Model (OAM) (Sudarsan et. al., 2004), an extension of the CPM, are developed in National Institute of Standards and Technology (NIST) to represent product information. The main component in the original Open Assembly Model (OAM) is the definition of "associations" among artifacts and their features. The spatial relationships in the assembly, and other connection and joint properties (i.e., degrees of freedom) of associated artifacts and their features are then defined. The original OAM model was in meta-level and not detailed; in this work, it has been detailed and extended by including details of associations among features and among artifacts. The detailing includes the spatial relationships in the assembly, and other connection and joint properties (i.e., degrees of freedom), of the associated artifacts and features.

Product structure is not the only product information considered in this work. Product functionality is another important factor to be considered in product development. Knowledge of product functions enables users to make intelligent decisions during the design process. In this study, functional information is considered throughout the product life cycle and includes the functional requirements or purpose of an artifact (e.g., transfer power), its functional input and output (e.g., rotational or translational mechanical energy, with attributes such as torque, force, angular and linear velocity), and the functional associations between artifacts and environment (e.g., transfer of motion, speed reduction, force transfer, etc.). Based on the functional requirements defined by customers, the main function is described and then decomposed into sub-functions. To perform the sub-level functions, appropriate artifacts are selected from a number of alternatives.

Behavior can be defined as the response of something (an artifact) to its environment. In this study, we treat artifact behavior as a result of interactions of the artifact with other artifacts in the assembly and with the environment, through a set of relevant functional relationships. A planetary gearbox is used as a case study to show how the functional/structural model can be implemented. A planetary gear has functional relationships (to transmit mechanical energy as force/velocity and torque/angular velocity) with other artifacts—the sun gear, the ring gear and the pin—which affect its behavior. The interactions of the planetary gear with the sun gear, the ring gear and the pin define the forces and moments on the free body diagram of the planetary gear. The planetary gear can then be designed based on physical laws, engineering formulas, and so forth, using "form" (material and geometry) information. All these physical laws and engineering formulas define the "Behavioral Model." In addition to the cases and conditions considered in the Behavioral Model, an artifact can have unintended behaviors because of unanticipated interactions with other artifacts and the environment, or because of a failure. Based on the behavior of the artifact, unintended behaviors (e.g., heat generation) might result, and new functions (e.g., heat removal) might need to be introduced to the model to overcome the unintended behaviors.

1.1.2 Product Information Exchange and Consistency Maintenance

The second goal is to define information exchange model with a tracing mechanism and a modification verification mechanism, based on the product information representation created above. The information-exchange model will provide mechanisms for capturing product information, storing it in a database, and allowing access to it. It will allow a user to edit, add, and transfer the information, employing a verification tool that will check the consistency of the modified information.

In this concept of product development, *consistency* becomes a very important aspect and it can be defined as the absence of contradiction (i.e., the ability to prove that a statement and its negative are both true) in a system. In addition, when modifying any part of the product information that is associated with other objects, the relevant modification(s) has to be approved by a verification tool. Along with consistency, another issue in an information model is *traceability*, which refers to “the capability for tracing artifacts along a set of chained operations, where these operations may be performed manually or with automated assistance” (Paige, 2008). In our case, traceability is provided by interrelating the objects (through functions, behaviors, features, etc.) in a way that enables one to follow the functions of the object from its main functional requirements to its sub-functions, and to its design rationale arguments. In regard to this, completeness becomes a very important issue. In this study, completeness is defined in terms of traceability, so there cannot be any deficiency or disconnect among the entities in the system. In other words, the information model has to be complete in terms of traceability of function, behavior, spatial relationships, and so forth, in order to support all information exchange activities.

1.2 Objectives of this Dissertation

In this study, a model for the representation of assembly-related product information, including product functions and behavior, is presented, to provide a mechanism for exchanging product information throughout the lifecycle of a product. The long-range goal is to develop a representation and exchange model for general product information, encompassing all product lifecycle activities, which can be applied to most electro-mechanical products. This will enable efficient collaboration among different stakeholders, reduce interoperability costs, and reduce product development time. To achieve the overall objective, the sub-objectives are as follows:

- to develop the representation of assembly-related product information, including information about the assembly structure, spatial and design relationships, and the connection/joint properties of associated artifacts and features,
- to define the assembly structure and associations, with mathematical characterizations to make the assembly model consistent, correct, and complete in terms of traceability,

- to develop functional and behavioral models that define the interrelationships among function, behavior and the form of an artifact throughout the product development stages. These interrelationships involve not only input/outputs (e.g., output speed and input speed), but also relations (e.g., associations between an artifact's spatial and design relations),

- to define interrelationships in the product information representation, in order to provide a basis for mechanisms to capture, store and access this information, and to enable transfer of the information using tools that will check the consistency of modified information.

- to define interrelationships among functional requirements, structure and behavior in a way that will provide a basis for a tracing mechanism that checks the consistency of the information and finds causes of failure by tracking through associations.

1.3 Organization of this Dissertation

Chapter 2 reviews the literature on (a) representation of product information including, structure, function and behavior (b) information modeling languages and (c) product information models and standards.

In chapter 3, the representation of the assembly-related product information in the modified Open Assembly Model (OAM) throughout the life cycle of the product is discussed. Spatial relationships in the assembly and the joint properties of associated artifacts and features are defined including the original and modified OAM. The interrelationships among different CPM and OAM classes, especially ones which are not directly related, are shown, to provide/maintain consistency in the product information model.

In chapter 4, a functional and behavioral representation model is developed to represent assembly-related product knowledge, such as functional requirements, the functional input and output of artifacts, and the functional associations between an artifact and the environment. A functional and behavioral model connects functions, behaviors and structure through the parts of artifacts, not only in regard to input/outputs, but also in regard to an artifact's spatial and design relations.

In chapter 5, a product information exchange model for assembly-related product information is developed, including a product information browser for browsing all objects. A tracing mechanism is defined for checking the consistency of the information and finding the causes of problem by tracking through functional associations and associations among artifacts. For modified information, a verification tool is introduced to check the consistency of the model.

Chapter 2

Review of Related Research

2.1 Product Function and Behavior Representation

In product representation, the extraction of geometry information from solid models has always been straightforward; however, it is necessary to identify semantic structures (i.e., features) for solid reasoning about a component's function. There are a number of methods and techniques for establishing a functional structure, and some efforts have been made to connect function and behavior with structure. However, a complete, correct and consistent product information representation is still lacking in literature. Therefore, it is necessary to develop a means of representation for function, behavior and artifact data models that will provide more correct and consistent product information to all agents in the product lifecycle.

Product functionality is one of the most important factors to be considered in product development. A *function* is defined as a system that has an objective to complete a predefined task by employing its input to deliver necessary output (Pahl & Beitz, 2007). In a technically complex system, the conversion of flows (energy, material and signals) between functions is best demonstrated by using a hierarchical structure to represent them. Many researchers have adapted and extended this input-output perspective (Gero, 1990; Gorti & Sriram, 1996; Kirschman & Fadel, 1998; Szykman et al., 1999; Otto & Wood, 2001).

Knowledge of product functions enables users to make intelligent decisions during the design of product modeling. The information model for assembly must include the functional and behavioral characteristics of component parts. There are a number of methods and techniques for establishing a function structure. A function block diagram is used to describe the overall function of an artifact, based on the flow of energy, material and signals, and to express the relationships between inputs and outputs (Pahl & Beitz, 2007)].

Campbell et al. (1999) developed a functional representation based on functional block diagrams, in which they show ports, or points of connectivity, with other components.

Function alone is not adequate for describing the multiple facets of product information. Usually, function is combined with behavior in product information modeling to ensure better decision making, where behavior represents the processes and principles that allow the function to be attained (Umeda et al., 1996; Chandrasekaran et al., 1993; Iwasaki et al., 1995). In other words, behavior essentially describes how a system behaves to fulfill the desired function. There have been numerous efforts by different researchers to synthesize the various facets of production information. As a result, many studies covering function-form-behavior models have been conducted

(Iwasaki & Chandrasekaran, 1992; Gorti & Sriram, 1996; Szykman et al., 2001; Roy et al., 2001). These studies allowed for the creation of different models for product information, in which different facets serve different purposes and have different influences on product design. In other words, they have been very useful in supporting product design.

Ullman (1993) observed the differences among part functions and proposed some definitions based on these differences, without taking into consideration behavioral interactions of the part at the geometry level. Therefore, Ullman's work is useful in the conceptual design phase, but not in detailed design phase. Chang et al. (2000) proposed an integrated system using form, function, and behavior-based (FFB-based) perspectives to fully describe any and all artifacts at any time during the design process, from conceptual design to detailed design. Oliver et al. (1997) developed Functional Flow Block Diagrams (FFBDs), in order to capture information about behavior from systems engineering. These diagrams are not computer-executable and have been augmented with input/output information.

Since the requirements are not complete at the beginning of almost every design process, requirement details are realized through the detailing of object descriptions--i.e., function is also detailed in design processes (Sudarsan et. al., 2005 and Takeda et. al., 1996). Takeda et al. (1996) termed the detailing of function a *functional evolution process*. Umeda et al. (1996) proposed the use of Function Behavior-State (FBS) diagrams to represent a function as an association of function and behaviors, rather than just either of them. FBS diagrams differentiate between the subjective parts of a design object (the functions and function-behavior relationship) and its objective parts (behaviors and states).

Some other efforts to link function, behavior and structure are listed below. Al-Hakim et al. (2000) proposed linking reliability with functional views, using graph theory to represent a product and the connections among its components, in order to trace any loss of functionality by easily visualizing the energy flow between components.

Brunetti and Golob (2000) suggested a feature-based representation scheme for capturing product semantics handled in the conceptual design phase. As information carriers to downstream applications, features are used to model the relationships among the requirements, functional descriptions and physical solutions of a product.

Lombeyda and Regli (1999) developed Conceptual Understanding and Prototyping (CUP), which allows users to specify a spatial layout of components and sub-assemblies and to establish their structural, functional and behavioral information. It also provides mechanisms for capturing textual information about the design intent and precedence during the conceptual design.

The Parametric Technology Corporation provides Pro/CONCEPT to support conceptual design, in addition to Pro/ENGINEER, but it does not maintain consistency between the model for the conceptual design phase and the model for the other design phases (Bronsvoort, 2004).

Additional studies have been carried out to establish function, behavior and form representations, as the foundation of a product information model that provides explicit linkages to ensure the consistency of product information in a distributed environment. Wang and Nnaji (2004) created a constrain-enabled UL-PLM model for this reason as well. The next step is to better define concepts for the description of knowledge, and then to give a basis for the systematization of knowledge provided by ontologies. Functional ontologies, including a device-centered ontology and a functional concept ontology, have been developed, focusing on the systematization of functional knowledge for design (Kitamura et. al., 2001).

2.2 Information Modeling Languages

Engineering design is conducted using different modeling languages, such as UML (Pulm and Lindemann, 2001), EXPRESS (ISO), and XML (Szykman et al., 1999; Rezayat, 2000). These languages are well suited for modeling a wide variety of physical processes and objects, owing to their common syntax and well-defined semantics. Moreover, these languages possess features that allow for excellent exchangeability, accessibility and interoperability of product information among diverse design groups.

The major information modeling languages in the literature are EXPRESS, UML, XML and OWL. There are also various standards developed by standard organizations and industry consortiums. These standards are domain specific based on XML, such as ebXML, STEPml, cXML, BizTalk, etc., (Eswaran, 2005). Peak et al. (2004) discuss efforts under way to make STEP based information models available through languages which are commonly used by more application developers, specifically XML and UML. They also present a vision and roadmap for integrating EXPRESS-based models with XML, UML, and other languages (e.g., OWL) to enable enhanced Product Lifecycle Management interoperability. Most researchers now prefer XML, UML or OWL over EXPRESS. Below are several reasons for this change in choice of standards:

- XML, UML and OWL are commonly used, and related resources (software/books) are broadly available, whereas EXPRESS is used by a very limited community.
- XML provides a standard syntax to represent structural data.
- XML, UML and OWL are better models for web applications. This makes distributed collaboration through the Internet easier and more convenient.

2.3 Product Information Representation Models and Standards

Product design requires complex interactions among system elements. In order to describe complex behaviors, it is necessary to explicitly model the use-environment. Shooter et al. proposed a model for the design of information flow (Shooter 2000). This model was further refined and resulted in the National Institute of Standards and Technology (NIST) Core Product Model (CPM) (Fenves 2001). The model provides a base-level product model that is open, non-proprietary, generic, extensible, independent of any product development process and capable of capturing the full engineering context commonly shared in product development. The CPM is intended to serve as a generic core representation for design information through the whole product development process. Specialized representations can be developed from it by deriving specialized classes.

Information models for function, assembly and behavior are critical in the conceptual development of a product, as well as during its evaluation. The NIST work on a core product model and its extension to an assembly model may serve as organizing principles for standards that may emerge in this area (Sudarsan, 2005; Baysal 2004, 2005). Zha et al. (2005) proposed a function-(environment-effect)-behavior-(principle-state)-form (FEEBPSF) framework based on the NIST core product model and its extensions, for modeling micro-electro-mechanical system (MEMS) products that apply the OESM (open embedded system model), which was developed to model information and knowledge for embedded MEMS design and development.

A similar effort is the ESPRIT-funded project MOKA (Methodology and tools Oriented to Knowledge-based engineering Applications). The MOKA modeling language is based on UML and is designed to represent engineering design knowledge at the user level for deployment in Knowledge Based Engineering applications (Sudarsan, 2005).

SysML is developed especially for the systems engineering domain based on UMLsic UML to cover the requirements, behavior, structure, and parametrics of structure and its relation to behavior (allocation). SysML reuses a subset of UML 2.0 diagrams and augments them with several new diagrams and modeling constructs that are used in systems modeling (Bock, 2004).

One of the most important open standards is STEP (Standard for the Exchange of Product Model Data), which was developed by the International Organization for Standardization (ISO) with the help of industrial consortiums such as PDES, Inc. (http://pdesinc.aticorp.org) and ProSTEP (http://www.prostep.de). The STEP (STEP – ISO 10303) consists of a family of standards defining a robust and time-tested methodology for describing product data throughout the lifecycle of a product. It provides a large body of standardized, strictly defined, highly dependable technical concepts. In the context of STEP, the product structure, geometry and part-related information are represented. STEP is widely used in Computer Aided Design (CAD) and Product Data Management (PDM) systems through application protocols (APs). APs describe the information model of a particular engineering or technical domain. For example, AP203 is the most used AP in CAD tools for configuration-controlled mechanical assembly design. APs and the resources used to develop them contain formally specified information models written in a language created especially for STEP, known as EXPRESS (Kemmerer, 1999; Pratt, 2001; Peak, 2002; Lubell, 2004).

For information exchange, there are some standards (EDI, SOAP and other specialized standards) for the exchange of data and information, but the most common one is XML. Specialized versions of these standards are: STEPml, a library of XML specifications based on the content models from the STEP standards; Product Data Markup Language (PDML) being developed as part of the Product Data Interoperability (PDI) project under the sponsorship of the Joint Electronic Commerce Program Office (JEPCO); PLMXML, a set of XML schemas serving as a transport protocol; and

Business Process Modeling Language, a meta-language for the modeling of business processes (Eswaran, 2005).

Current product information standards (e.g. IGES, STEP etc.) emphasize the structural and static relationships of entities. Variant relations among geometric entities (constraints) cannot be represented.

2.4 Summary

Information models for function, assembly and behavior are critical for the conceptual development of a product and for its evaluation. There has been much research conducted in this area to represent and interrelate all aspects of product information. Even though many good studies have been successful in relating some aspects of the product information, they do not relate function, behavior and structure in a comprehensive and consistent way.

In this work, the NIST CPM and OAM models are extended to represent product information. The Functional basis method (Pahl & Beitz, 2007; Stone & Wood, 2000) is adapted for functional structure definition. For function and flow classification, the NIST functional taxonomy (Hirtz et al., 2002) Szykman et al., 2000; Stone & Wood, 2000) is used in this study as it is used consistently throughout the literature. The screw theory application for assembly constraints from Whitney (2004) and Adams (1998) is applied to mathematically define assembly constraints and relations in the assembly through the degrees of freedom property. Also, interrelations among some of the packages/classes in the OAM model (which are similar to some of the parts in the STEP [i.e. Kinematic

Structure in ISO 10303-105]) are discussed, and consistency rules among those classes are defined for consistent product information representation and exchange.

Chapter 3

Representation of Assembly-Related Product Information

3.1 Product Information Representation

In this chapter, assembly-related product information is represented, including assembly structure, spatial and design relationships and the connection/joint properties of associated artifacts and features. Then, the assembly structure and associations in the product model are defined through a mathematical characterization, to ensure that the representation of assembly structure is consistent, correct and complete in terms of traceability. This product information representation will be a foundation for exchanging product information throughout the lifecycle of the product.

Since, the main issue is interoperability and exchanging product information among product life cycle activities, in National Institute of Standards and Technology (NIST) Core Product Model (CPM) and Open Assembly Model (OAM) (Sudarsan et. al., 2004), an extension of the CPM, are developed to overcome interoperability issues. The original OAM model was in meta-level and not detailed; in this work, it has been detailed and extended by including details of associations among features and among artifacts. The detailing includes the spatial relationships in the assembly, and other connection and joint properties (i.e., degrees of freedom), of the associated artifacts and features. The extension includes interrelationships among different classes of artifacts–especially the ones which are not directly related–, in order to provide and maintain consistency in the product information model. The original model is also extended by adding new classes (i.e., parametric assembly constraint realize etc.) and consistency rules, which will be discussed in chapter 5. This will enable efficient collaboration among different stakeholders and reduce the interoperability costs as well as product development time.

In this chapter, after the OAM and its modifications are described in detail, a gearbox design problem is discussed to show the value-added information that we are providing in the OAM to realize a seamless integration between product information and product design throughout all phases of the product's model fabrication.

3.1.1 The Open Assembly Model

The Open Assembly Model (OAM), developed in NIST, is extensible; it currently provides tolerance representation and propagation, representation of kinematics, and engineering analysis at the system level (Rachuri, 2003). The assembly information model emphasizes the nature and information requirements for part features and assembly relationships. It uses the model data structures of ISO 10303, informally known as the STandard for the Exchange of Product model data (STEP). The main difference between the OAM and many other available standards is that the assembly model is not at the end of the product design; instead, it evolves from an incomplete, preliminary form to a complete model as the design progresses from early to detailed design phases. The model starts with customer-specified functions and functional requirements. On completion of the design, the OAM databases contain detailed information regarding function, behavior, form/structure, kinematics, assembly and tolerance for the entire product. A brief discussion of the OAM and modifications are given in this chapter; for more information about the OAM, please refer to Appendix 1 (Rachuri, 2003). It uses the model data structures of ISO 10303, informallay known as the Standard for the Exchange of the Product model data (STEP).

Figure 3.1 shows the main schema of the modified Open Assembly Model. The added associations, classes and package are shown with thicker lines in the figure. The schema incorporates information about assembly relationships and component composition; the former is represented by the class AssemblyAssociation; the latter is modeled using part-relationships. The class AssemblyAssociation represents the component assembly relationship of an assembly, and consists of the aggregation of one or more Artifact Associations. The ArtifactAssociation class represents the assembly relationship between two or more artifacts. An Assembly is decomposed into subassemblies and parts. A Part is the lowest level component. Each assembly component (whether a sub-assembly or part) is made up of one or more features, represented in the model by OAMFeature. The Assembly and Part classes are subclasses of the CPM Artifact class, and the OAMFeature is a subclass of the CPM Feature class.

ArtifactAssociation is the generalization of the following classes: PositionOrientation, RelativeMotion and Connection. PositionOrientation represents the relative position and orientation between two or more artifacts that are not physically connected and describes constraints on the relative position and orientation between them. RelativeMotion represents the relative motions between two or more artifacts that are not physically connected and describes the constraints on the relative motions between them.

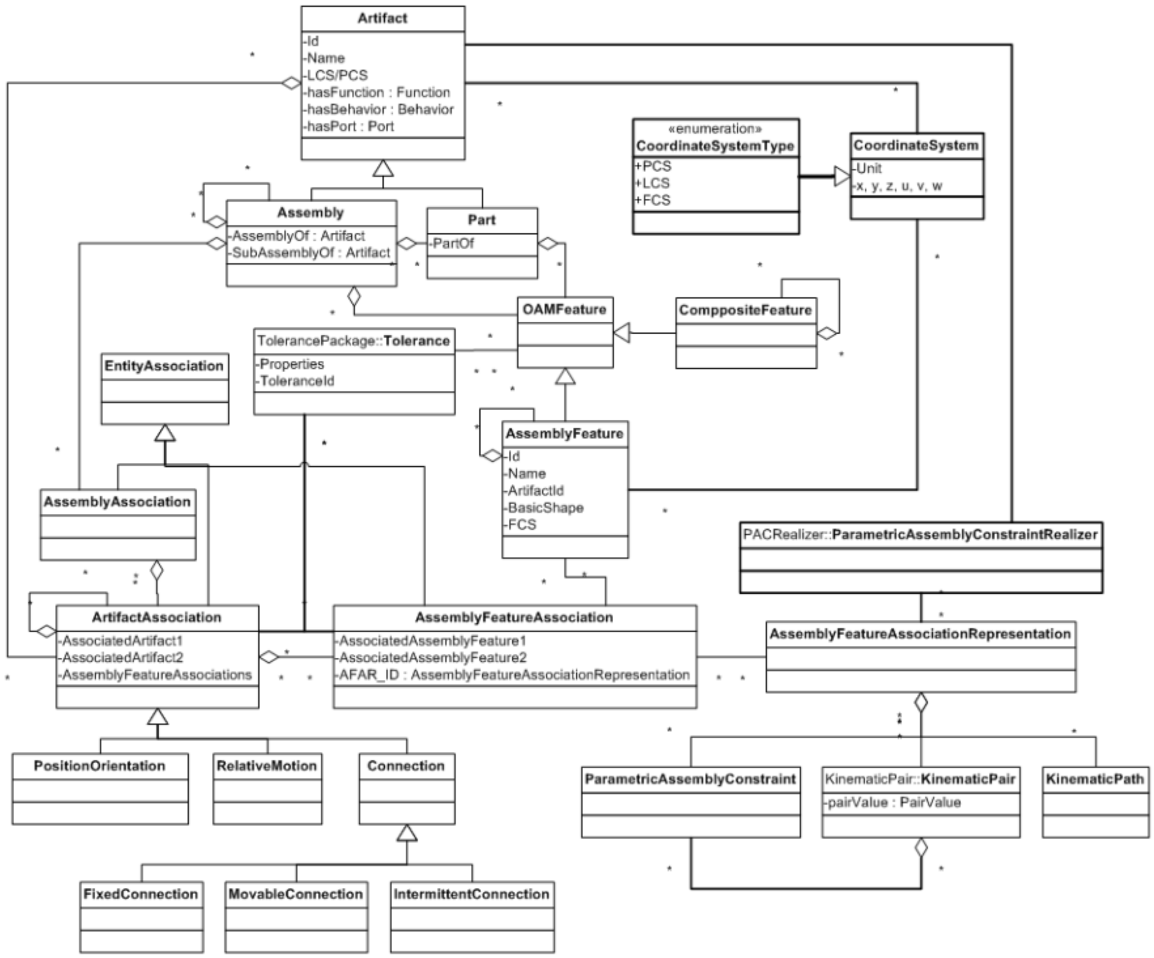


Figure 3.1: Assembly-related Associations in the Modified Open Assembly Model

Connection represents the connection between artifacts that are physically connected. Connection is further specialized as FixedConnection, MovableConnection, or IntermittentConnection. FixedConnection represents a connection in which the participating artifacts are physically connected and describes the type and/or properties of the fixed joints. MovableConnection represents a connection in which the participating artifacts are physically connected but movable with respect to one another and describes the type and/or properties of kinematic joints. IntermittentConnection represents a connection in which the participating artifacts are physically connected only intermittently. Detailed relationships between Connection and parametric assembly constraints are defined in this work and given in Section 3.2.5.

KinematicPair defines the kinematic constraints between two adjacent artifacts (links) at a joint (Figure 3.2). The kinematic structure schema in ISO 10303-105 defines the kinematic structure of a mechanical product in terms of links, pairs, and joints.

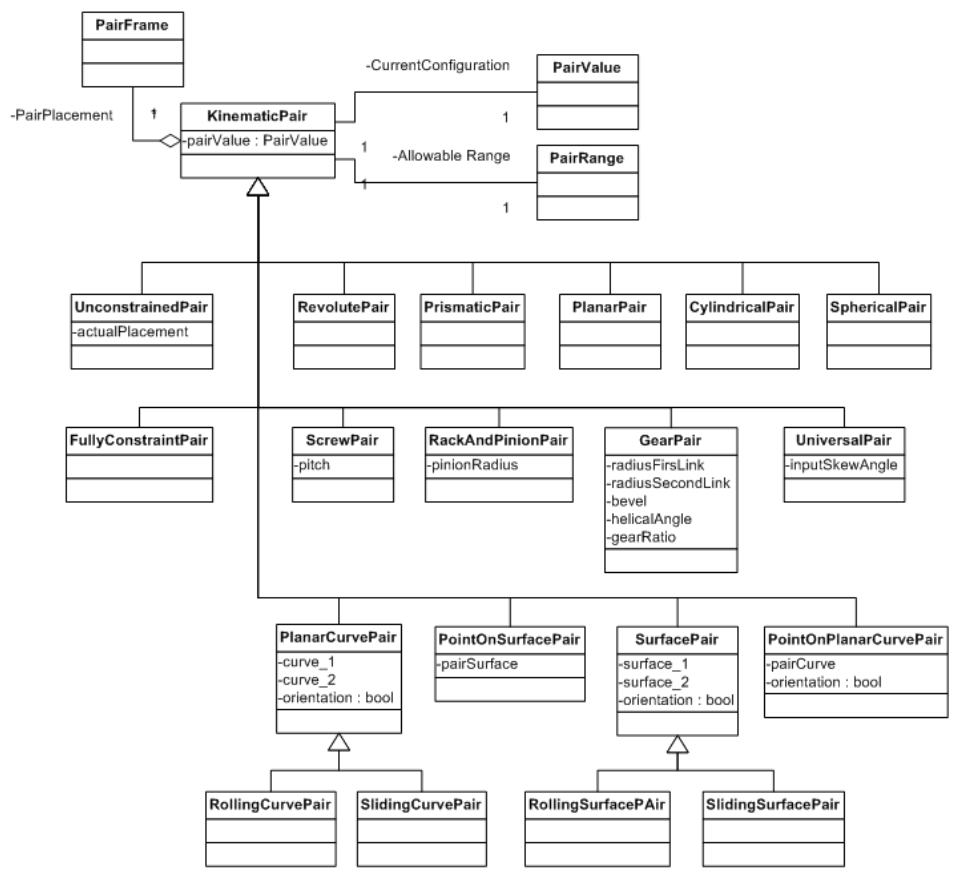


Figure 3.2: KinematicPair information in the OAM

The OAMFeature consists of tolerance information, represented by the class Tolerance and the sub-classes CompositeFeature and AssemblyFeature. CompositeFeature represents a complex feature that can be decomposed into multiple simple features. AssemblyFeature represents 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 cylindrical surface 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, screws and nuts, planes spheres, cones, and toruses as assembly features. Dimensional and geometric tolerance information is stored in tolerance objects, as defined in the tolerance classes in Figure 3.3.

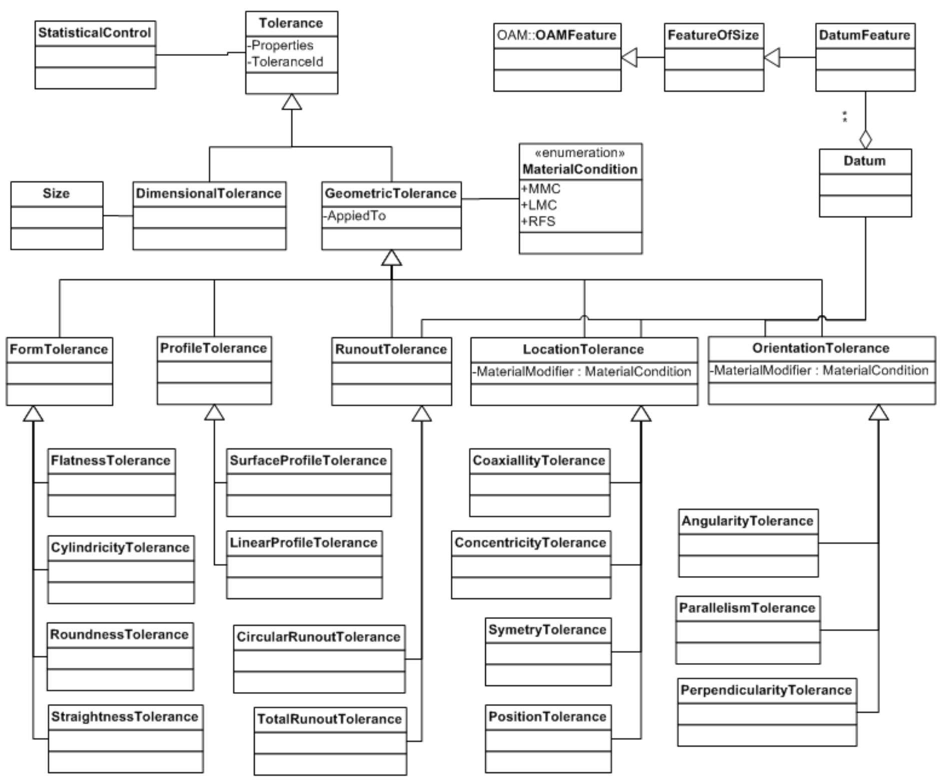


Figure 3.3: Tolerance in the Open Assembly Model

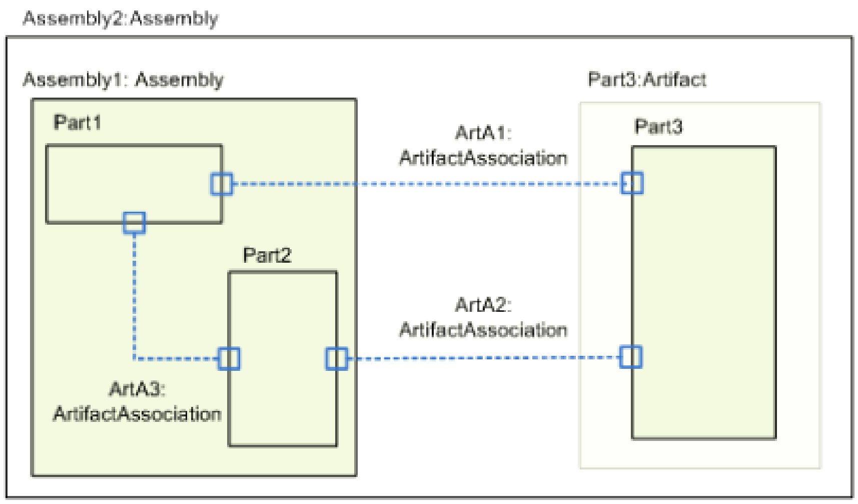
3.1.2 Representation of Associations in the Modified OAM

The associations in the OAM are described at three different levels: the levels of assembly, artifact (part) and assembly feature. Figure 3.1 incorporates information about assembly relationships and component composition. As mentioned above, the class AssemblyAssociation represents the component assembly relationship of an assembly. It is the aggregation of one or more Artifact Associations. The ArtifactAssociation class represents the assembly relationship between one or more artifacts. The class AssemblyFeatureAssociation represents the association between the mating assembly features through which relevant artifacts are associated. The class ArtifactAssociation is the aggregation of the AssemblyFeatureAssociation.

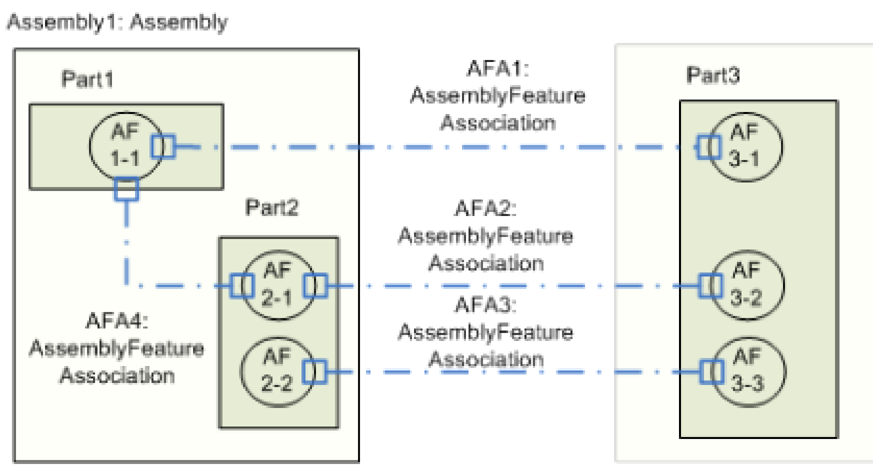
In the conceptual design phase, system level artifacts (main assembly and major parts) are defined with incomplete information. For example, a solution, which will become an artifact, is defined without any information on its behavior or form/structure at the beginning of the design. The part-level information (i.e., basic shape, type of part such as gear etc.,) is introduced in the preliminary design phase, and then the remaining information (detailed geometry, material etc.) is provided during the detailed design and other phases. From the conceptual design stage to the detailed design phases, the associations are specified one by one, beginning with the artifact associations (in the conceptual and preliminary design phases) to the assembly feature associations and kinematic relations (in the preliminary and detailed design phases). After the artifacts are designed in the detailed design phase, the assembly features and associations between them are then defined as detailed representations of the artifact associations. In Figure 3.4, the association levels in the assembly are shown. The first-level association, the assembly association, includes all of the artifact associations in an assembly. In the second-level association, artifact associations are established between artifacts: ArtifactAssociation1 between Part1 and Part3 (Figure 3.4a). In the sub-level of ArtifactAssociation2, there are two assembly associations: AssemblyFeatureAssociation1 between AssemblyFeature2-1 of Part2, and Assembly Feature3-1 of Part3, and AssemblyFeatureAssociation2 between AssemblyFeature2-2 of Part2 and AssemblyFeature3-2 of Part3 (Figure 3.4b).

In other words, assembly associations are the upper level associations which define the relationships at the assembly/subassembly level. At the next level, artifact associations are defined between artifacts at the part level. Finally, assembly feature associations are defined at the assembly feature level. At the same time, in the modified OAM, assembly feature associations aggregate to artifact associations, and artifact associations aggregate to assembly associations (as shown in Table 3.1).

Since the relations of the three abstraction level of associations (classes) are defined in the modified OAM, it is important to connect the geometry information of those classes with rules and constraints. In chapter 5, information regarding geometry and other constraints relating to the classes will be discussed in detail.



(a) Assembly and Artifact Associations



(b) Assembly Feature Associations

Figure 3.4: Representation of Associations in the Modified OAM

Table 3.1: Associations in Different Abstract Levels

Assembly Associations	Artifact Associations	Assembly Feature Associations
AA1 = {ArtA3}	ArtA3 = Part1 and Part2	AFA4 = AF1-1 and AF2-1
AA2 = {AA1, ArtA1, ArtA2} = {ArtA3, ArtA1, ArtA2}	ArtA1 = Part1 and Part3	AFA1 = AF1-1 and AF3-1
	ArtA2 = Part2 and Part3	AFA2 = AF2-1 and AF3-2 AFA3 = AF2-2 and AF3-3

3.1.3 Representation of Geometry Information

One of the main issues with product information representation in the modified OAM is the representation of the geometry information of the entities in relevant classes in a consistent way. The basis of geometry information in the modified OAM is the ISO-10303 (STEP) standard. The necessary information for the modified OAM may be extracted from the STEP data structure. Other design information related to the function, behavior, design rationale, etc., is built up within the model. Geometry information in STEP is very extensive and is associated with other standards and parts in STEP. The modified OAM model cannot be fully populated and tested without a geometry information structure. Extracting the required information from STEP to the modified OAM is a complicated and time-consuming process. The total mapping of STEP entities (30,000 definitions for transfer from a CAD to another CAD tool (Ray, 2002)) to the modified OAM is not possible at present. Therefore, in this work, instead of representing detailed geometry information (by mapping STEP to UML) in the OAM, geometry information is defined in three abstraction levels (Table 3.2): (1) the basic geometry information of an artifact, with the position and orientation information of an artifact within its assembly, (2) functional features (i.e., assembly features and their interrelationships, as well as type and basic shape information), and (3) detailed geometry information of all features in an artifact.

The first two levels of geometry information are enough to satisfy the requirements for the OAM classes/objects for the assembly/tolerance related purposes of this study.

Table 3.2: The Geometry Information Abstraction Levels for the Gear Example

Abstraction Levels	Description
1st level: (Artifact)	Artifact:Pin1: BasicShape.Cylinder (diameter=,length=) Artifact: Gear1: BasicShape.Gear (hole_diameter = , hole_depth = , pitch_dia, etc.)
2nd level: (Assembly Feature)	AF1:PinCyl1: BasicGeomEntity.Cylindrical : Centre (x, y, z), Radius = , length = , AF2:GearHole1: BasicGeomEntity.Cylindrical : Centre(x, y, z), Radius = , length = ,
3rd level: (Detailed)	Detailed geometry information of the artifacts (points, edges, surfaces, volumes, etc.,)

The model can then be used without requiring detailed geometry information. But these two levels of information are not sufficient for a complete representation of the geometry information of a product, which may be required to support all other product life cycle tools. This makes the detailed (3rd level) geometry information an important issue which needs to be solved.

Artifacts and features in the assembly structure are represented as in a Constructive Solid Geometry (CSG) representation which makes the information model be mathematically traceable by defining relative position and orientation of each feature/artifact to other features/artifacts. In this CSG-like representation, parts have local coordinate systems (LCS), and functional features have feature coordinate systems (FCS), which are defined according to their position and orientation relative to the LCS. Therefore, relative position/orientation of any feature/artifact with respect to any other feature/artifact in the product assembly can be determined mathematically. Primitives are basic shapes (cylinder, sphere, etc.) with geometric information (radius, length, etc.), as shown in Figure 3.5.

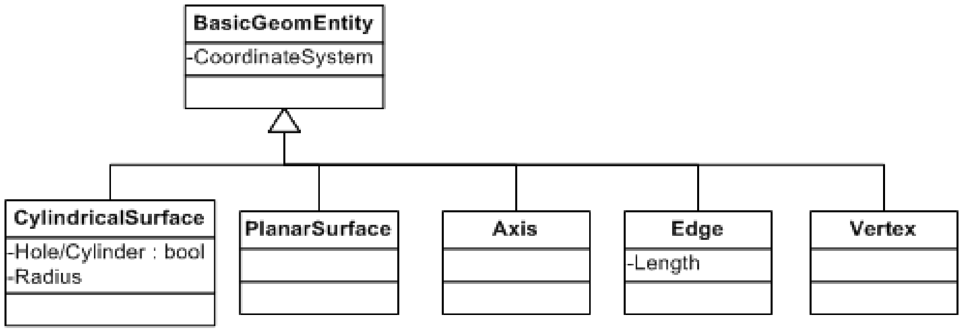


Figure 3.5: Basic Geometric Entities in CSG like Representation

3.2 Definitions for the Consistency of Assembly Associations

It is necessary to describe the interrelations among objects, which are instances of artifact, function, behavior, feature, etc., classes in application level for a particular product, in such a way that they enable tracing/navigating objects through associations/interrelations in the assembly. There cannot be any deficiency or disconnection among these objects. In other words, the information model has to be complete in terms of the traceability of function, behavior, and assembly associations, in order to support all information modeling and exchange activities. For this reason, structure and associations are defined mathematically. First, associations in the assembly are defined. Every artifact and feature has its own position relative to a coordinate system, and the orientations and features of artifacts, and other information nodes, are first defined with transformation matrices, which enable the calculation of the positions and orientations of entities with respect to each other. Several transformations can easily be chained by multiplying the corresponding matrices. Second, connections/assembly constraints/joints between parts and between features are defined based on degrees of freedom, by applying screw theory (chapter 4). Third, relations among these association classes are defined at different levels and from different perspectives (i.e., assembly constraints, kinematic pairs, geometric tolerances, etc.). All of these matrix-based definitions for positions/orientations, connections, and assembly constraints, along with functional inputs/outputs and the behavioral model (which are described in chapter 4), are then utilized to develop a traceable product information model mathematically.

3.2.1 The Relative Position and Orientation of Assembly Features and Artifacts

In this section, we define the position orientation of assembly features according to an artifact's local coordinate system. Since the assembly, artifact and feature associations are the key elements in our model, the relative positions and orientations of the parts and features and artifacts (based on the artifact's local coordinate system [LCS] and the feature coordinate system [FCS]) are defined by transformation matrices and stored in the association classes in the modified OAM model. In this study, these transformation matrices (T) can be a combination of artifact and assembly feature associations (Whitney, 2004).

$$T = \begin{bmatrix} r_{11} & r_{12} & r_{13} & p_x \\ r_{21} & r_{22} & r_{23} & p_y \\ r_{31} & r_{32} & r_{33} & p_z \\ 0 & 0 & 0 & 1 \end{bmatrix} \Rightarrow T = \begin{bmatrix} R & p \\ 0^T & 1 \end{bmatrix} \quad p = \begin{bmatrix} x \\ y \\ z \end{bmatrix} \qquad (3.1)$$

In equation (3.1), p gives the translational transformation in x, y, and z directions, and R gives the orientation. Equation (3.2) gives the combined transformation matrix between Artifacts A and B (in the ArtifactAssociation class in the modified OAM model):

$$T_{PartA-PartB} = T_{PartA-FeatureA1}\ T_{FeatureA1-FeatureB1} T_{FeatureB1-PartB} \tag{3.2}$$

where $T_{PartA-FeatureA1}$ the relative position and orientation of Feature A_1 in Part A (in the AssemblyFeature class), $T_{FeatureA1-FeatureB1}$ between Feature A_1 and Feature B_1 (in the AssemblyFeatureAssciation class) ,and $T_{FeatureB1-PartB}$ Feature B_1 in Part B. Thus, the result of any alteration in the position or orientation of any artifact or feature can then be modified accordingly. When many parts are joined this way, one can navigate from part to part by following the transformation frames. The relative position and orientation of Gear1 with respect to Pin1 through their associated assembly features is defined by the combined transformation matrix as following;

$$T_{Pin1-Gear1} = T_{Pin1-Pin1AF}\ T_{Pin1AF-Gear1AF}\ T_{Gear1AF-Gear1} \tag{3.3}$$

Then, the connection (joint) properties of associated artifacts and assembly features are defined by the frames of the assembly features (or links) and the degree of freedom of the connection, and they are stored in the AssemblyFeature, AssemblyFeatureAssociation, Connection, KinematicPair classes of the OAM. Figure 3.6 shows assembly features, their individual relative position and orientation (P/O) to the local coordinate system (LCS) of the artifact (transformation matrices - $T_{AF.FCS-Art.LCS}$). It also shows ports, which are special features to include association information about functional association, assembly feature associations and assembly feature association representation including details about kinematic pairs (e.g. gear pair).

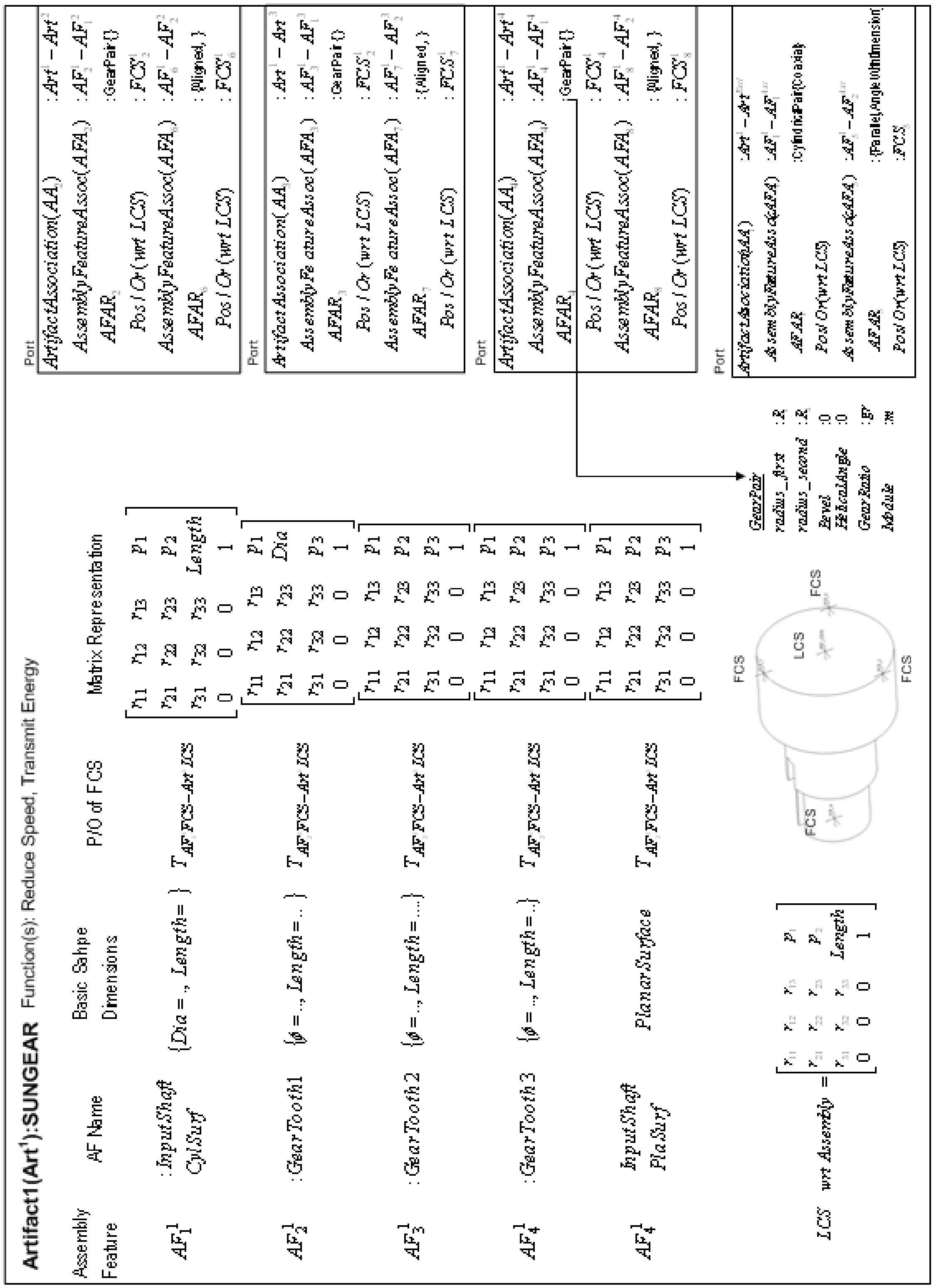


Figure 3.6: Assembly Feature Level Information of Sungear

3.2.2 Degrees of Freedom and Effects on Consistency

Artifact association, assembly feature association, assembly feature association representation, connection type, kinematic pair and parametric assembly constraint are all related to each other. In addition to the specialized relation between some classes, there is a need for common parameter to provide consistency among them. Therefore, in this study, *degrees of freedom* is used as a common parameter to control the consistency among different classes. *Degrees of freedom* (d-o-f) defines the relative motion capability of one artifact with respect to another. An unconstrained artifact in space has six degrees of freedom: three translational and three rotational. When two artifacts are associated through their assembly features with constraints, the relative motion capability will be reduced to a value between 0 and 5, based on the type of constraint. There might be more than one constraint between two artifacts through different assembly feature associations with various d-o-f's. In that case, the d-o-f's of all assembly feature associations between the same artifacts are combined to determine the d-o-f of the artifact association. The type of connection between two artifacts is related to the degrees of freedom of the artifact association. When the degrees of freedom are zero, the Connection type is fixed; if not, it is moveable or intermittent. On the other hand, when product information is formed, the type of connection might be defined as fixed in the Connection class, but if the parametric assembly constraint is modified for some reason, it changes the degrees of freedom from zero to some value between 1 and 6. In such a situation, the connection is no longer fixed. Therefore, this shows the importance of relating the Connection, KinematicPair, ParametricAssemblyConstraint, and relevant classes by d-o-f common parameter.

3.2.3 Relationships among Assembly Features

After we define the assembly features (the basic shapes and basic geometric entities defined as primitives in CSG) and establish the position/orientation of a given assembly feature relative to the local coordinate system of an artifact, as well as the positions/orientations of artifacts relative to other artifacts, the next step is to define the assembly constraints (degrees of freedom, [d-o-f], etc.) between components. The motions (translational and rotational) related to the functional association are derived from spatial relationships and design requirements.

Assembly constraint types are given in Table 3.3 (i.e., align, parallel, etc.). These types of assembly constraints have different available motions that affect the relative motions of various parts. The assembly associations between artifacts' features can result in a fixed or moveable connection between artifacts. A fixed connection, which has zero degrees of freedom, can be permanent or detachable. Examples of fixed connections are welding, soldering, brazing, adhesive bonding, and interference fits, while examples of detachable types include common mechanical fasteners like bolt-nuts and screws, and clearance fits. When the degree of freedom between two artifacts is other than "0," as a result of combinations of assembly constraints between the features of the artifacts, then that connection between the artifacts is called "moveable."

Assembly constraints are defined in the OAM using the ParametricAssembly Constraint class, which is derived using the Assembly_Geometric_ Constraint entity presented in ISO 10303-109. This is a super-type of the Binary_Assembly _Constraints, which are also a subset of relevant geometric constraints like the parallel_assembly_constraints, which are, in turn, a subset of the parallel_geometric_constraint.

Table 3.3: Assembly Constraints in the OAM and STEP Standard

ParametricAssemblyConstraint in the OAM	Binary_Assembly _Constraints in STEP
Parallel (line or plane)	Parallel Assembly Constraint
ParallelWithDimension	
SurfaceDistanceWithDimension	Surface Distance Assembly Constraint with Dimension
AngleWithDimension	Angle_assembly_constraint_with_dimension
Perpendicular	Perpendicular_assembly_constraint
Incidence	Incidence_assembly_constraint
Coaxial	Coaxial_assembly_constraint
Tangent	Tangent_assembly_constraint
Fixed	Fixed

3.2.4 Parametric Assembly Constraint Realizers

Associations between artifacts are related to the parametric assembly constraints between associated assembly features. Although parametric assembly constraints give information about relative motion capability, it is necessary to define how they are constrained physically, and by what. Therefore, assembly constraints must be realized by physical entities (i.e., bolts, welding, etc.). The Parametric Assembly Constraint Realizer (PACR) is defined to realize parametric assembly constraints by defining special artifacts, joining processes, pair mesh, and physical constraints like friction and gravity. At the same time, it relates parametric assembly constraints to function and behavior. So we can say that the intended behavior of the parametric assembly constraint is realized by the ParametricAssemblyConstraintRealizer (Figure 3.7). If something happens related to the parametric assembly constraint's associated behavior, we can track down the problem through the ParametricAssemblyConstraintRealizer by checking whether or not the artifacts are working properly.

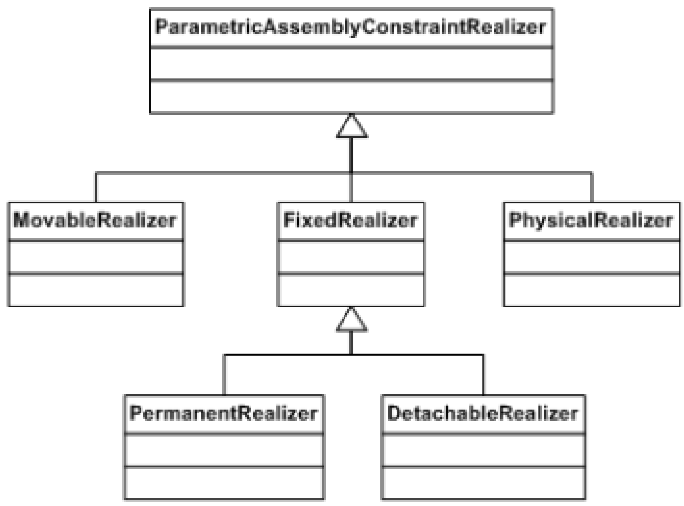


Figure 3.7: Parametric Assembly Constraint Realizer UML Class

The PACR is also related to the disassembly modeling of the product. It is easier to find out which artifacts hold the assembly together and to know what will happen if that artifact, which is a ParametricAssemblyConstraintRealizer, is removed. For example, a bolt may have "Hold" function, and separating of two parts by taking out the connecting bolt from the assembly may cause to fail its secondary function "Seal". So, containing liquid (i.e. oil) may spill. More dangerously, a hazardous material can leak and cause dramatic consequences.

Even though the Connector, which is defined as specialized artifact, class is introduced via the CORE model, in this study it is defined as a part of the DetachableRealizer class in the ParametricAssemblyConstraintRealizer package (as shown in Figure 3.7 and Table 3.4). It is not only defined as a special artifact in the CORE model, but also as a connecting process, which is classified based on the movability and permanence of the connection (Table 3.3). In addition, PhysicalRealizer defines other physical entities and laws (e.g., friction, gravity, magnetism etc.,) so as to realize the parametric assembly constraints. In this study, ParametricAssemblyConstraintRealizer examples are given in Table 3.4.

Table 3.4: Some Examples of Parametric Assembly Constraint Realizers

Parametric Assembly Constraint Realizers			
Fixed Realizers		Movable Realizers	Physical Constraints
Permanent Realizers (Joining Processes)	Detachable Realizers (Joining Connectors)		
Welding Soldering Brazing Riveting Gluing	Bolt-Nut-Washer Screw TaperFit KeyFit PinFit Clutch-Disc	GearMesh BeltMesh ScrewMesh Journal Bearing RubberRing RollFit SpringFit	Gravity Friction Magnetism Geometry-Interference

3.2.5 Effect of Tolerances and Fit Types

Fit type and tolerance are used to determine the degrees of freedom. The tolerance fit type will also affect the degrees of freedom by physically preventing motion, even when there is a Cylindrical Pair defined between the assembly features. When the tolerance value of the diameter of the hole or the cylinder is modified, and the fit type involves interference, then this connection turns into a fixed connection, or vice versa. Therefore, relevant consistency rules have to be defined. The ParametricAssemblyConstraint is related to geometric tolerances through the TolerancedAssemblyConstraint subtype of the ParametricAssemblyConstraint.

3.2.6 Relationships among Connection, Assembly Constraints and Kinematic-Pair

Connection, ParametricAssemblyConstraints and KinematicPair classes use related / matching product information, so it is necessary to define how they are related. Then, consistency rules will be developed for the reliable product information model. Relationships among the Connection, ParametricAssemblyConstraints and KinematicPair classes are shown in Figure 3.8. As we mentioned in section 3.2.2, the Connection type of artifact association depends on the combined degrees of freedom, a value which is calculated from the entire group of assembly constraints between assembly features of the same artifact. At the same time, type of kinematic pair may be defined by the combination of assembly constraints between assembly features of theses artifact. For example, the coaxiality assembly constraint between two cylindrical surfaces (one a hole, the other a cylinder) gives us a cylindrical pair. Both have the same degree of freedom in translational and rotational movability along the axis. Here are the requirements;

- the basic shape of both assembly features has to be a cylindrical surface;
- one must be a hole, other must be a cylinder;
- the axes of the cylindrical surfaces must be aligned (coaxial);
- the tolerance type must be clearance;

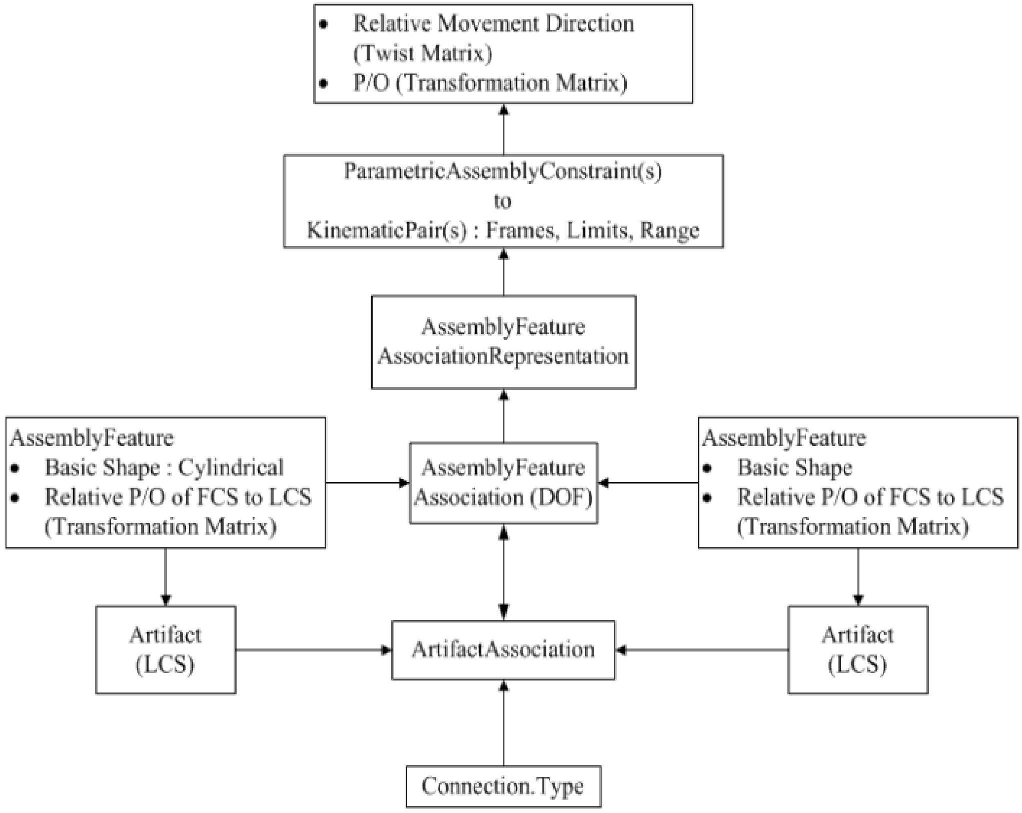


Figure 3.8: The OAM Classes that are Using Same Information

For example, the relation between a KinematicPair frame and an artifact's (and assembly feature's) position is given in Figure 3.9.

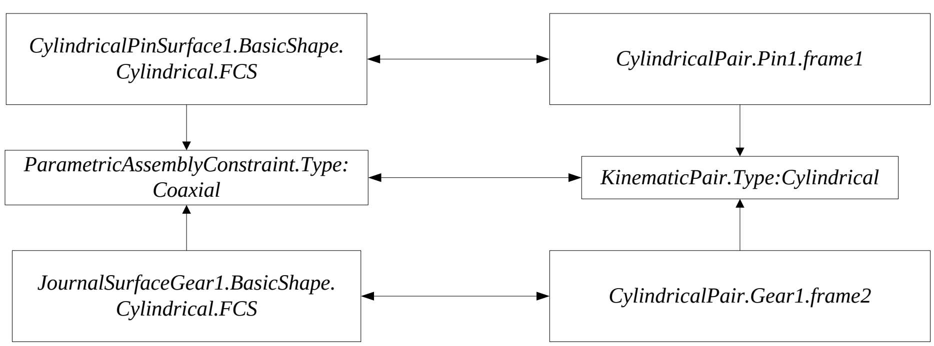


Figure 3.9: The relationship between AssemblyConstraint and KinematicPair

Here, the CylindricalPair class (a sub-class of the KinematicPair class) has two transfer_item's (artifacts: Pin1 and Gear1) and these transfer_item's have the "frame (x, y, z)" attribute for the position information of the artifacts. The same position information for both artifacts needs to be stored in the "centre (x, y, z)" attribute (for cylindrical surfaces) of the assembly features (CylindricalPinSurface1 and JournalSurfaceGear1).

3.2.7 Relationships between Assembly Constraints and Geometric Tolerances

In regard to the artifacts Pin1 and Gear1, there is a design requirement for the concentricity of the gear journal and the pin. In order to define these types of relationships, whether they be between the different artifacts in an assembly or between three levels of geometric information for the same artifact, it is necessary to establish rules/constraints." As per our design requirements, we now have to establish the equivalence between the concentricity information in the geometry tolerance of Pin1 and also establish the kinematic pair relationships between Gear1 and Pin1, using constraints. For concentricity, the relationship is:

ParametricAssemblyConstraint.AssemblyFeature.BasicGeomEntity.Cylindrical.Centre ⇔ Tolerance.GeometricTolerance.CrossReferenced.Location.Concentric.Datum.df.Feature.Axis .

That is, the AssemblyConstraint class for this association has two cylindrical assembly features (CylindricalPinSurface1 and JournalSurfaceGear1) with center information. This position information needs to be equivalent to the "axis" information in the concentric geometry tolerance for the same assembly features.

3.3 Information Flow through Product Design Stages in the Modified OAM

In this section, flow of assembly related product information through product design stages in the product information representation model (OAM) is discussed and illustrated by the gearbox example. So, it will show how product information is populated in what stages of product design. Figure 3.10 explains how the OAM populates product information within product development.

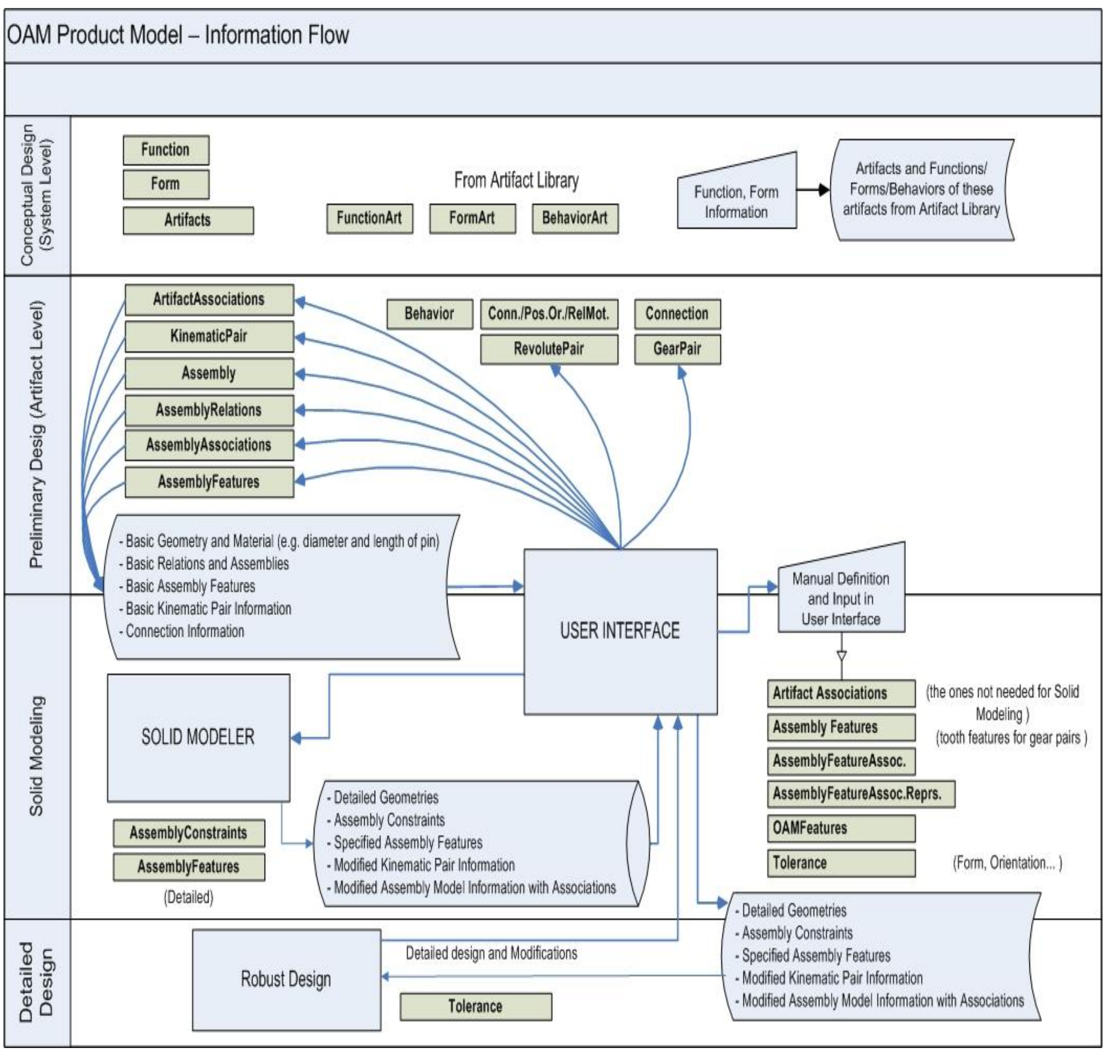


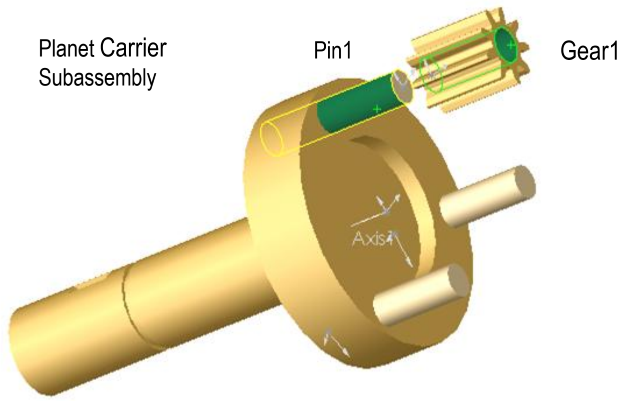
Figure 3.10: The OAM Information Flow

The OAM is based on associations in the assembly at different levels. Starting from the conceptual design stage and moving to the detailed design stages, we specify the associations one by one, starting with assembly and artifact associations, then progressing to assembly feature associations and kinematic relations. In the conceptual design stage, we know what some of the major functional artifacts are going to be, so we can specify associations between them. When we specify an association between two artifacts, we also need to specify some extra information about that association.

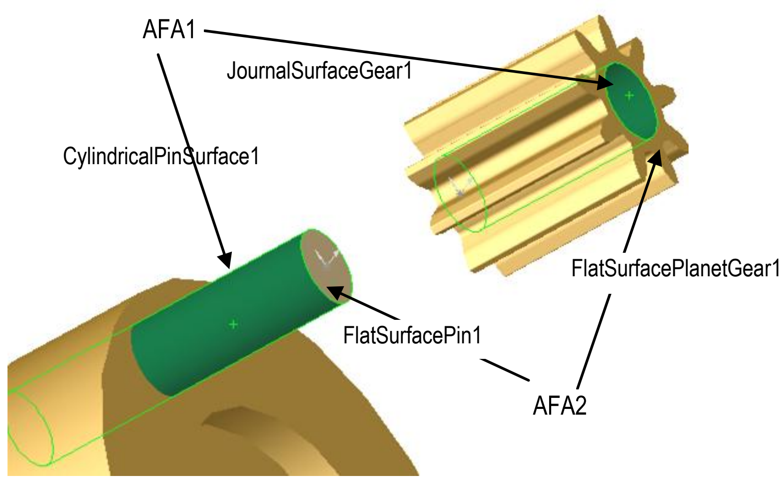
3.3.1 Relationships among OAM Classes

In this section, we are going to show how the modified OAM handles associations, and geometry tolerance information for a subassembly of a model planetary gearbox. The geometry information for associations needs to be defined at different levels. The necessary geometry information in the artifacts, associations and other classes can then be used for the life cycles of other products life cycle, i.e., tolerance analysis, assembly planning, and so forth. In the modified OAM, in the conceptual design stage, the first sets of data entered into the system are for the major functional artifacts (e.g., the planetary gear carrier, planetary gear, etc. as shown in Figure 3.11-a). In later stages, the data for other artifacts (e.g., the pin, output shaft) and the associations among all those artifacts (as shown in Figure 3.11-b) are entered at the assembly and artifact levels. For the gearbox, the artifacts are the Planet Gears, the Pins, the Planet Gear Carrier Subassembly (including the shaft, pins and gears), and the Planet Carrier Subassembly (including the shaft and pins).

Once we have designed the artifacts, the associations among these artifacts are defined from the assembly to the part level. To begin with, the associations among the artifacts, according to assembly associations, are defined. Then, the more detailed associations between assembly features are established, based on assembly constraints (Table 3.5).



(a)



(b)

Figure 3.11: The Pin1 and the Gear1 Association at the Assembly Feature Level

For example, when a Concentricity assembly constraint between the cylindrical surfaces of Pin1 and Gear1 and a "coincident" assembly constraint between the flat surfaces of Pin1 (CylindricalPinSurface1) and Gear1 (JournalSurfaceGear1) (Figure 3.11) are defined, it means, first, that there is an association between the Planet Carrier Subassembly and Gear1 at the assembly level and, therefore, an association between the artifacts Pin1 and Gear1, and, second, that there are associations (AF1 and AF2) between the assembly features (as shown in Figure 3.12), which are described as mating features in the assembly constraints.

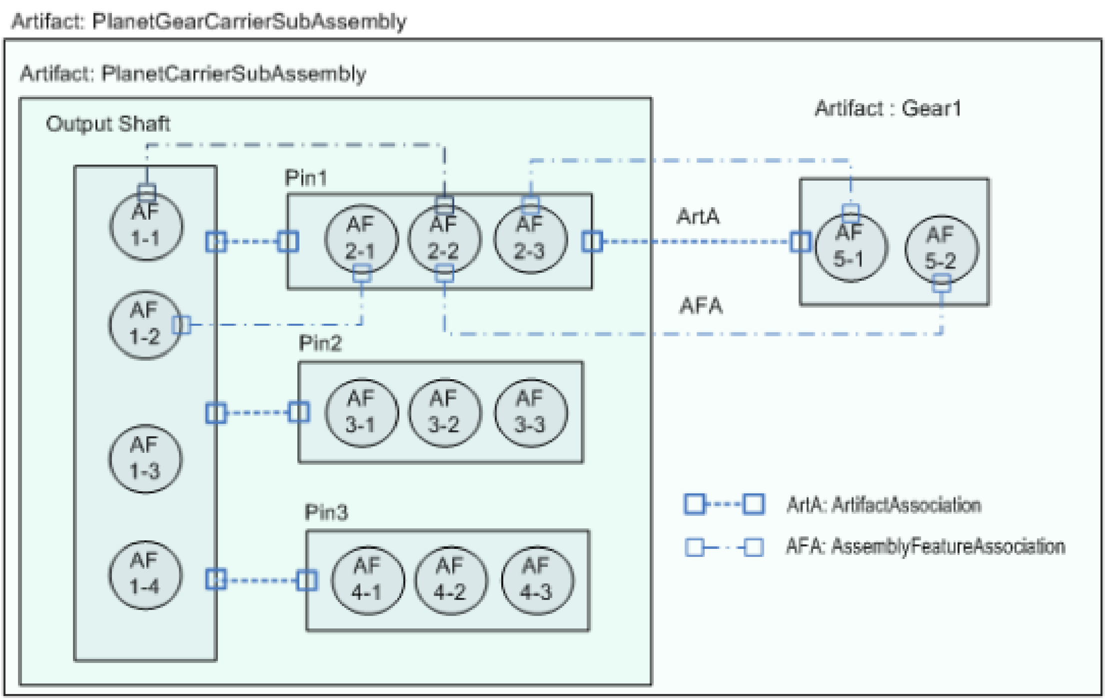


Figure 3.12: Three Different Associations in the Planet Gear Carrier Subassembly

Table 3.5: Association Levels in the Planet Carrier Sub Assembly Assembly

Association Level	Associated Elements
Assembly Level	Artifact Associations in Planet Carrier Sub Assembly
Artifact Level	Pin1 – Gear1
Assembly Feature Level	1) CylindricalPinSurface1 (AF 2-2) ⇔ JournalSurfaceGear1 (AF 5-2)
	2) FlatSurfacePin1 (AF 2-3) ⇔ FlatSurfacePlanetGear1 (AF 5-1)

At this point, the associations among entities at the assembly, artifact and assembly feature levels are defined in the modified OAM, and the relationships among relevant objects are provided by rules/constraints. For example, the relation between the Kinematicpair.CylindricalPair.frame and the assembly feature's coordinate system is defined as

CylindricalPinSurface1.BasicShape.Cylindrical.Centre = CylindricalPair.Pin1.Frame.z

JournalSurfaceGear1.BasicShape.Cylindrical.Centre = CylindricalPair.Gear1.Frame.z

The information about the PlanetaryGearPin1 (Pin1), the relevant assembly features, the kinematic pair and the artifact association also includes kinematic and tolerance information in relevant (KinematicPair and Tolerance) classes. Consequently, this information can be applied to any product lifecycle tool (e.g., tolerance analysis tools) for particular purposes (tolerance analysis, assembly planning, etc.). In this example, the kinematic pair of Gear1 and Pin1 is a cylindrical pair which has PairValue and PairRange information (based on STEP), and the local coordinate systems of the parts in the "frame" attribute. The local coordinate for Pin 1 is stored in frame1 $\{x_5, y_5, z_5\}$, while that for Gear1 is stored in frame2 $\{u_5, v_5, w_5\}$, as shown in Figure 3.13.

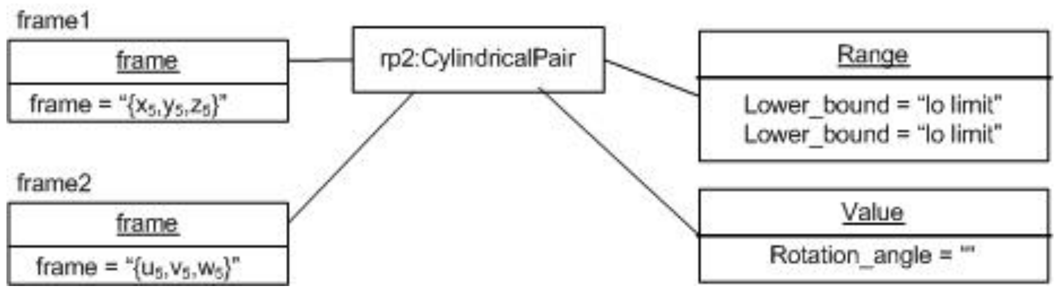


Figure 3.13: Representation of the KinematicPair information of the Gear1-Pin1 Assembly

Figure 3.14 shows the horizontal dimensions of a three-part assembly (Gear1, Pin1 and the output shaft), the gap (between Gear1 and Output shaft), and their related surfaces. In the figure, for number '21, the first number (2) represents a specific part (Gear1), while the second number (1) represents the surface on that part. The dimensional chain and tolerance chain are defined as in equations (3.4) and (3.5).

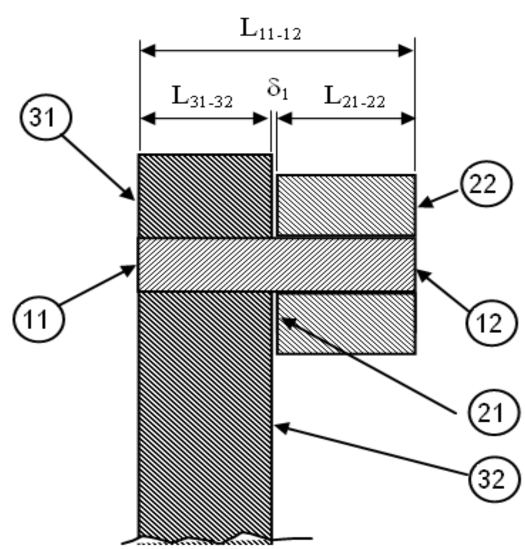


Figure 3.14: Dimensions and related surfaces in the PlanetGearCarrier subassembly

The dimensional chain is

$$\delta_1 = L_{11-12} - L_{21-22} - L_{31-32} \tag{3.4}$$

The tolerance chain for δ_1 is

$$T_{\delta_1} = t_{11-12} + t_{21-22} + t_{31-32} \tag{3.5}$$

In the downstream tolerance analysis of the PlanetGearCarrier subassembly, the required information about the sizes, positions and orientations of Gear1 and Pin1, and other assembly features, is extracted from the relevant classes (Artifact and AssemblyFeature) in the modified OAM. The local coordinate system (LCS) gives the positions of the artifacts, and the feature coordinate system (FCS) gives the center of the assembly feature (Figure 3.15). For the positions of surfaces in the horizontal (x) direction, we need to use only the *x* component of the FCS (for the PlanarSurface-32, the coordinates are (0, 0, 0); for the PlanarSurface-31, the coordinate is (12.7, 0, 0)). After extracting this dimensional information only in the *x*-direction, one can perform the 1-D stack-up analysis using equations (3.4) and (3.5).

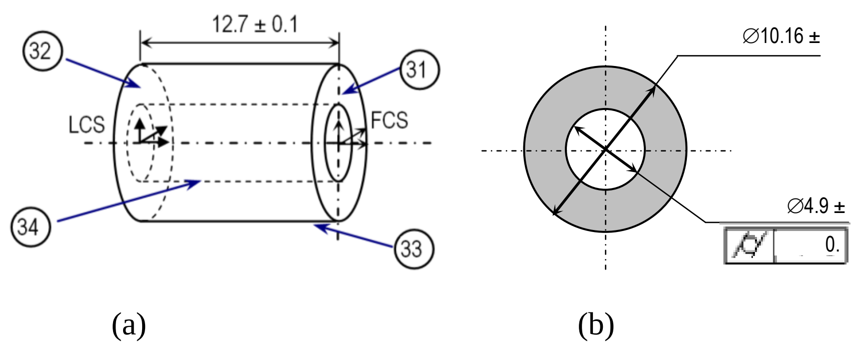


Figure 3.15: Tolerances and Associated Features of Gear 1

The artifact "Gear1" has two geometric tolerances and one dimensional tolerance (as shown in Figure 3.15) which are represented in the OAM objects (instances) as shown in Figure 3.16. The tolerance information stored in the OAM objects consists of tolerance type, tolerance zone and reference datum for referenced tolerances.

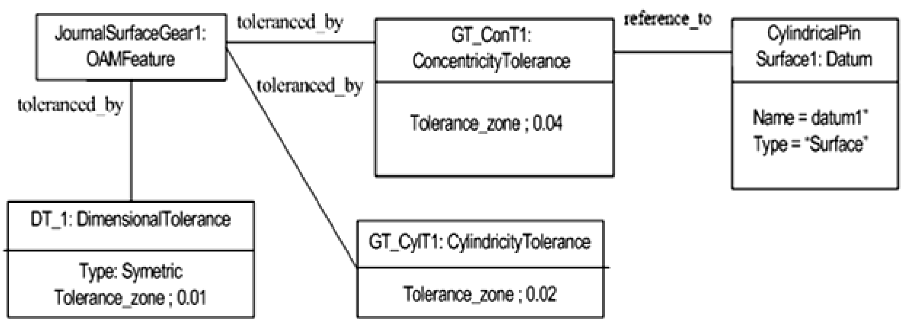


Figure 3.16: Tolerance information for JournalSurfaceGear1 in the OAM

Similarly, CylindricalPinSurface1:OAMFeature has cylindricity geometric tolerance and dimensional tolerance shown in Figure 3.17.

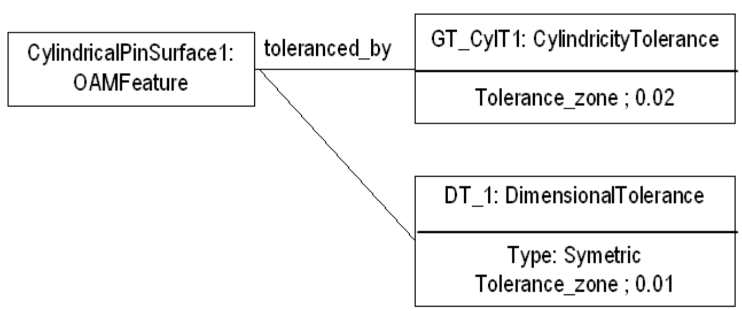


Figure 3.17: Planet Tolerance information for Cylindrical Surface of Pin1 in the OAM

Now, assembly related product information for the example of a gearbox represented by OAM objects (instances) are populated in tables. Artifacts (i.e. parts and assemblies in Figure 3.18 are listed in Table 3.6.

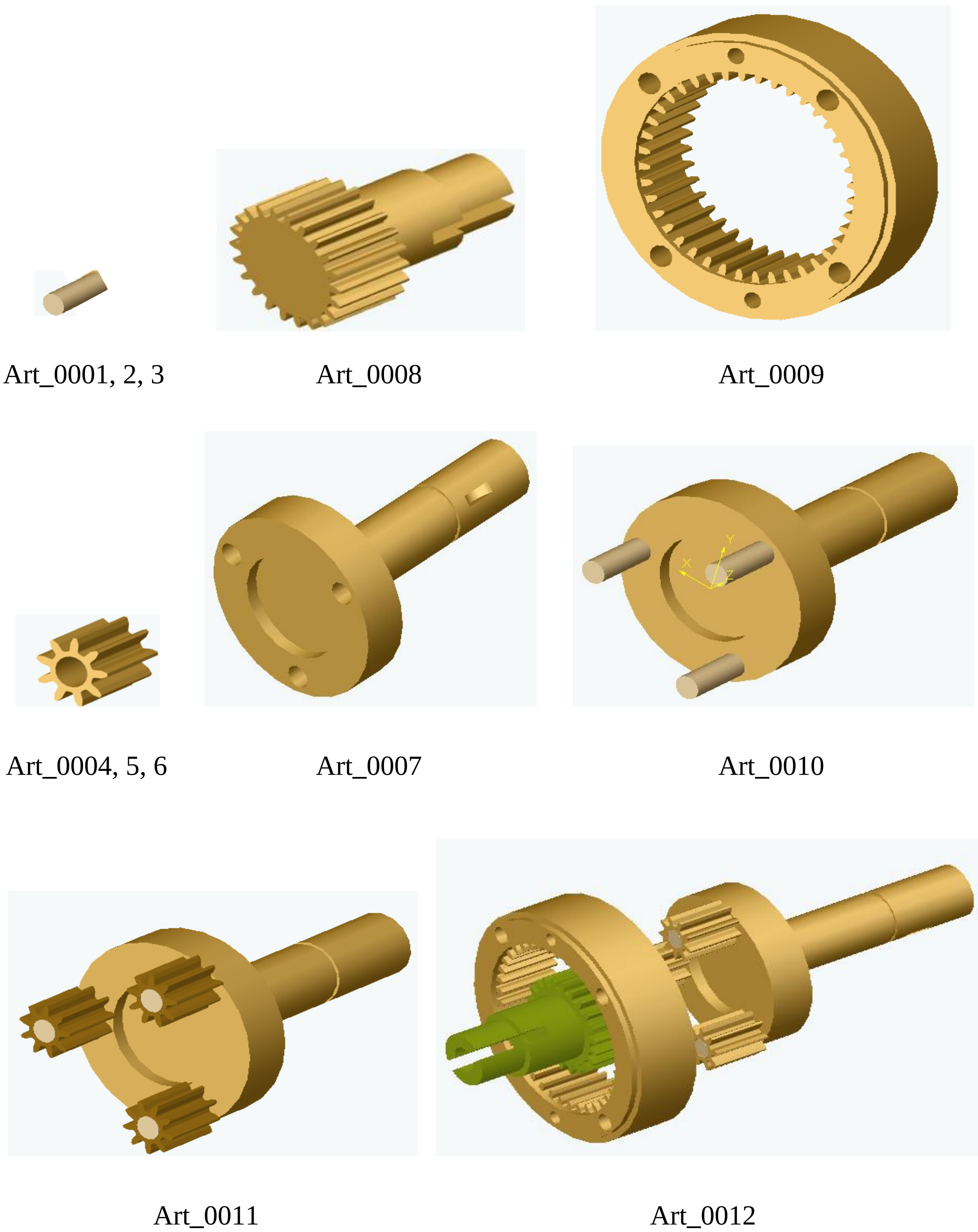


Figure 3.18: 3D Models of Artifacts (Parts and Subassemblies)

AssemblyFeatures extracted from Part/Assembly are given in Table 3.7 and Table 3.8 (OAMFeature). Associations between the assembly features (AssemblyFeatureAssociation - AFA) are listed in Table 3.9. For each AFA, an AssemblyFeatureAssociationRepresentation (AFAR) subclass is defined which are specialized into ParametricAssemblyConstraint (shown in Figure 3.19 and listed in Table 3.9 and Table 3.10), KinematicPair (Table 3.11 and Table 3.12) and KinematicPath. An artifact association can be of three types: connection, position-orientation and relative motion. Combination of the ParametricAssemblyConstraints between particular two artifacts, define the artifact associations (Table 3.13). There are some associations among artifacts that are not directly connected to each other (i.e, PositionOrientation in Table 3.15 and RelativeMotion). These define the artifact association similar to the manner it is done in Connection. The physically connected artifact associations are described in the Connection (Table 3.14) with the connection type (moveable, fixed or intermittent), related assembly features, assembly constraints, and the kinematic pair information.

In other words, from the information given ParametricAssemblyConstraints, the associated assembly features are determined as individual assembly features, unlike in the ArtifactAssociation table. AssemblyFeatureAssociation and Connection/PositionOrientation/ RelativeMotion classes aggregate the ArtifactAssociation, and in turn ArtifactAssociation(s) in an assembly aggregates the AssemblyAssociation. AssemblyAssociations are defined by the aggregation of artifact associations in an assembly (Table 3.15).

The stages in which information is entered into the OAM product model are (see **Error! Reference source not found.**):

1. Conceptual Design (customer needs, specifications, functional design),
2. Preliminary Design (part design, assembly tree, assemblies),
3. Solid Modeling of parts and assemblies,
4. Completing information by user interface
5. Detailed Design (modified part/assembly design) and
6. Analysis (functionality, tolerance, assemblability)
7. Default.

Table 3.6: The Artifacts (Parts and Assemblies) in Planenatry Gearbox

Artifact

Id	Name	Defin.	Group/ Type	Function_Id	Behavior_Id	Form Id	Requires
0001	PlanetGearPin1 [2]	Part [2]	Locator[2]	0009[1]	-	0001	0007
0002	PlanetGearPin2 [2]	Part[2]	Locator[2]	0009[1]	-	0001	0007
0003	PlanetGearPin3 [2]	Part[2]	Locator[2]	0009[1]	-	0001	0007
0004	PlanetGear1 [2]	Part[2]	P_T[2]	0008[1]	-	0002	0001, 0007
0005	PlanetGear2 [2]	Part[2]	P_T[2]	0008[1]	-	0002	0002, 0007
0006	PlanetGear3 [2]	Part[2]	P_T[2]	0008[1]	-	0002	0003, 0007
0007	OutputShaft [2]	Part[2]	P_T[2]	0007[1]	-	0003	UnDef
0008	SunGear [2]	Part[2]	P_T[2]	0006[1]	-	0004	UnDef
0009	RingGear [2]	Part[2]	P_T[2]	0005[1]	-	0005	UnDef
0010	PlanetCarrierSubAssembly [2]	SA[2]	P_T, Locator[2]	0004[2]	-	0007	0001, 2 3, 7
0011	PlanetGearCarrierSubassembly [2]	SA[2]	P_T[2]	0003[2]	-	0008	0004,5, 6,7,10
0012	PlanetGearSubassembly [2]	SA[2]	P_T[2]	0001,0002[2]	-	0009	0008,9, 11
0013	Planetary_GearBox_Subassem. [2]	SA[2]	P_T[2]	0001,0002[1],	0001[2]	0010[1]	0012, UnDef

Table 3.7: Assembly Features Extracted from Part/Assembly Files

AssemblyFeature

Id	Name	Artifact	Definition / Parameters
0001 [2]	PinHole3 [2,3]	0007 [2]	D= D_{ph3} , L= L_{ph3} [2,3]
0002 [2]	PinHole4 [2,3]	0007 [2]	D= D_{ph4} , L= L_{ph4} [2,3]
0003 [2]	PinHole5 [2,3]	0007 [2]	D= D_{ph5} , L= L_{ph5} [2,3]
0008 [2]	PinCylinder3 [2,3]	0001 [2]	L= depth of Pinhole_1 [2,3]
0009 [2]	PinCylinder4 [2,3]	0002 [2]	L= depth of Pinhole_2 [2,3]
0010 [2]	PinCylinder5 [2,3]	0003 [2]	L= depth of Pinhole_3 [2,3]
0004 [2]	PinCylinder6 [2,3]	0001 [2]	L= length of GearJournal_Surface_1 [2,3]
0005 [2]	PinCylinder7 [2,3]	0002 [2]	L= length of GearJournal_Surface_2 [2,3]
0006 [2]	PinCylinder8 [2,3]	0003 [2]	L= length of GearJournal_Surface_3 [2,3]
0007 [2]	Cylinder [2,3]	0010 [2]	D= D_{011} , L= L_{0011} [2,3]
0011 [2]	GearJournal_Surface_1 [2,3]	0004 [2]	D= D_{gc1} , L= L_{gc1} [2,3]
0012 [2]	GearJournal_Surface_2 [2,3]	0005 [2]	D= D_{gc2} , L= L_{gc2} [2,3]
0013 [2]	GearJournal_Surface_3 [2,3]	0006 [2]	D= D_{gc3} , L= L_{gc3} [2,3]
0014 [2]	Teeth_1 [2,4]	0008 [2]	Teethform_1 [2,4]
0015 [2]	Teeth_2 [2,4]	0008 [2]	Teethform_1 [2,4]
0016 [2]	Teeth_3 [2,4]	0008 [2]	Teethform_1 [2,4]
0017 [2]	Teeth_4 [2,4]	0009 [2]	Teethform_1 [2,4]
0018 [2]	Teeth_5 [2,4]	0009 [2]	Teethform_1 [2,4]
0019 [2]	Teeth_6 [2,4]	0009 [2]	Teethform_1 [2,4]
0020 [2]	Teeth_7 [2,4]	0004 [2]	Teethform_1 [2,4]
0021 [2]	Teeth_8 [2,4]	0004 [2]	Teethform_1 [2,4]
0022 [2]	Teeth_9 [2,4]	0005 [2]	Teethform_1 [2,4]
0023 [2]	Teeth_11 [2,4]	0006 [2]	Teethform_1 [2,4]
0024 [2]	Teeth_10 [2,4]	0005 [2]	Teethform_1 [2,4]
0025 [2]	Teeth_12 [2,4]	0006 [2]	Teethform_1 [2,4]
0026 [2]	Cylinder [2,3]	0010 [2]	L= width of bearing_Journal [2,3]
0027 [2]	Bearing_Journal [2,3]	Bearing [2]	D= D_{bj1} , W= W_{bj1} [2,3]

Table 3.8: OAMFeatures with Tolerance in the Gearbox Assembly

OAMFeature (Only Toleranced Features are Listed)

Id	Feature Name	Artifact	Tol_Id
OAMF_1	EndSurface1	Art_0008	GT_1, GT_2, DT_1
OAMF_2	Sungear_teeth	Art_0008	DT_2
OAMF_3	Shank	Art_0008	GT_3, DT_3, DT_4
OAMF_4	inputshaft	Art_0008	GT_4, DT_5
OAMF_5	PinHole6:AF	Art_0004	GT_5, DT_6
OAMF_6	GearCylinder1	Art_0004	DT_7, DT_8
OAMF_7	rimsurface	Art_0009	GT_6
OAMF_8	GearTeethHole	Art_0009	DT_9
OAMF_9	PinHole1	Art_0009	DT_10
OAMF_10	PinHole2	Art_0009	DT_10
OAMF_11	EndSurface2	Art_0007	GT_7, GT_8
OAMF_12	outputShaftShank	Art_0007	GT_9
OAMF_13	Keyway	Art_0007	

Table 3.9: Assembly Feature Associations

AssemblyFeatureAssociation (AFA)

Id	Art_1	Art_2	AF_1	AF_2	AFAR
AFA_1 [2]	0007 [2]	0001 [2]	PinHole3 [2,3]	PinCylinder3 [2,3]	AFAR_1 [2]
AFA_2 [2]	0007 [2]	0002 [2]	PinHole4 [2,3]	PinCylinder4 [2,3]	AFAR_2 [2]
AFA_3 [2]	0007 [2]	0003 [2]	PinHole5 [2,3]	PinCylinder5 [2,3]	AFAR_3 [2]
AFA_4 [2]	0001 [2]	0004 [2]	PinCylinder6 [2,3]	GearJournal_Surface_1 [2,3]	AFAR_4 [2]
AFA_5 [2]	0002 [2]	0005 [2]	PinCylinder7 [2,3]	GearJournal_Surface_2 [2,3]	AFAR_5 [2]
AFA_6 [2]	0003 [2]	0006 [2]	PinCylinder8 [2,3]	GearJournal_Surface_3 [2,3]	AFAR_6 [2]
AFA_7 [2]	0004 [2]	0008 [2]	Teeth_7 [2,4]	teeth_1 [2,4]	AFAR_7 [2]
AFA_8 [2]	0005 [2]	0008 [2]	Teeth_9 [2,4]	teeth_2 [2,4]	AFAR_8 [2]
AFA_9 [2]	0006 [2]	0008 [2]	Teeth_11 [2,4]	teeth_3 [2,4]	AFAR_9 [2]
AFA_10 [2]	0004 [2]	0009 [2]	Teeth_8 [2,4]	teeth_4 [2,4]	AFAR_10 [2]
AFA_11 [2]	0005 [2]	0009 [2]	Teeth_10 [2,4]	teeth_5 [2,4]	AFAR_11 [2]
AFA_12 [2]	0006 [2]	0009 [2]	Teeth_12 [2,4]	teeth_6 [2,4]	AFAR_12 [2]
AFA_13 [2]	0010 [2]	Bearing [2]	Cylinder [2,3]	Journal [2,3]	AFAR_13 [2]

Table 3.10: ParametricAssemblyConstraints in the Gearbox Assembly

ParametricAssemblyConstraint

Id	Artifacts	Assembly Features (AF)	Type
AC_1 [3]	OutputShaft, PlanetGearPin_1 [3]	Cylindrical_PinHoleSurface_1, CylindricalPinSurface_1 [3]	Coaxial [3]
AC_2 [3]	OutputShaft, PlanetGearPin_1 [3]	Flat_PinHoleSurface_1, Flat_PinSurface_1 [3]	Coincid. /Parallel [3]
AC_3 [3]	OutputShaft, PlanetGearPin2 [3]	Cylindrical_PinHoleSurface_2, CylindricalPinSurface_2 [3]	Coaxial [3]
AC_4 [3]	OutputShaft, PlanetGearPin2 [3]	Flat_PinHoleSurface_2, Flat_PinSurface_2 [3]	Coincid. /Parallel [3]
AC_5 [3]	OutputShaft, PlanetGearPin3 [3]	Cylindrical_PinHoleSurface_3, CylindricalPinSurface_3 [3]	Coaxial [3]
AC_6 [3]	OutputShaft, PlanetGearPin3 [3]	Flat_PinHoleSurface_3, Flat_PinSurface_3 [3]	Coincid. /Parallel [3]
AC_7 [3]	PlanetGearPin_1, PlanetGear_1 [3]	Cylindrical_PinHoleSurface_1, JournalSurface_Gear_1 [3]	Coaxial [3]
AC_8 [3]	PlanetGearPin_1, PlanetGear_1 [3]	FlatSurface_OutputShaft, FlatSurface_PlanetGear_1 [3]	Coincid. /Parallel [3]
AC_9 [3]	PlanetGearPin_2, PlanetGear_2 [3]	Cylindrical_PinHoleSurface_2, JournalSurface_Gear_2 [3]	Coaxial [3]
AC_10 [3]	PlanetGearPin_2, PlanetGear_2 [3]	FlatSurface_OutputShaft, FlatSurface_PlanetGear_2 [3]	Coincid. /Parallel [3]
AC_11 [3]	PlanetGearPin_3, PlanetGear_3 [3]	Cylindrical_PinHoleSurface_3, JournalSurface_Gear_3 [3]	Coaxial [3]
AC_12 [3]	PlanetGearPin_3, PlanetGear_3 [3]	FlatSurface_OutputShaft, FlatSurface_PlanetGear_3 [3]	Coincid. /Parallel [3]
AC_13 [3]	OutputShaft, Sungear [3]	CylindricalSurface_OutputShaft, JournalSurface_SunGear [3]	Coaxial [3]
AC_14 [3]	OutputShaft, RingGear [3]	CylindricalSurface_OutputShaft, JournalSurface_SunGear [3]	Coaxial [3]

Table 3.11: KinematicPair (CylindricalPair) for the Gearbox Assembly

KinematicPair

Id	Name	Transform_item_1	Transform_item_2	PairValue	Frame1	Frame2
CP_1	CP_1[2]	UnknownSupport 2	Art_00082 (Sungear)	Rotation_ angle = θ1 [2,3]	{x1 y1 z1 }[2]	{u1 v1 w1 }[2,3]
CP_2	CP_2[2]	Art_0004 2 (Planetgear1)	Art_0001 (PGPin1) [2]	Rotation_ angle = θ2 [2,3]	{x1 y1 z1 }[2]	{u1 v1 w1 }[2,3]
CP_3	CP_3[2]	Art_0005 2 (Planetgear2)	Art_0002 (PGPin2) [2]	Rotation_ angle = θ3 [2,3]	{x1 y1 z1 }[2]	{u1 v1 w1 }[2,3]
CP_4	CP_4[2]	Art_0006 2 (Planetgear3)	Art_0003 (PGPin3) [2]	Rotation_ angle = θ4 [2,3]	{x1 y1 z1 }[2]	{u1 v1 w1 }[2,3]
CP_5	CP_5[2]	Art_0010 [2] (PlanetCarrierSubA)	Bearing) [2]	Rotation_ angle = θ5 [2,3]	{x1 y1 z1 }[2]	{u1 v1 w1 }[2,3]

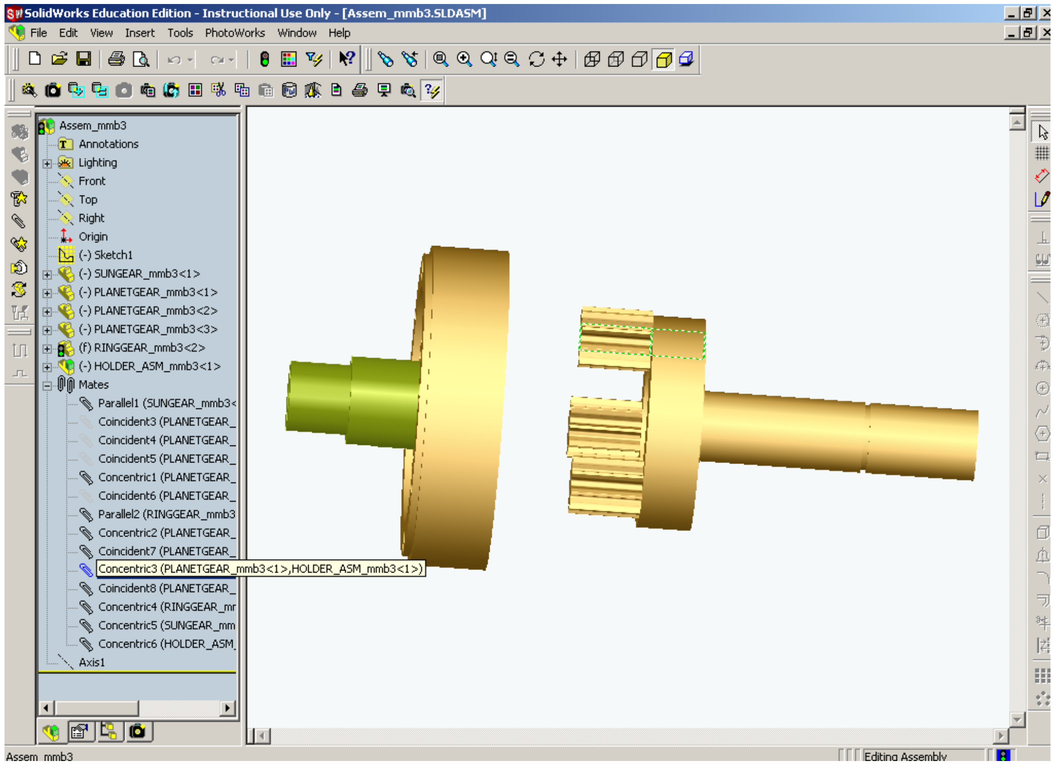


Figure 3.19: Assembly Constraints (Mates) for the Gearbox Assembly (in SolidWorks)

Table 3.12: KinematicPair (GearPair) for the Gearbox Assembly

GearPair

Id	Transform_item_1	Transform_item_2	radius_first_link	radius_second_link	gear_ratio	Bevel (plane_angle measure)	helical_angle plane_angle_measure
GP_1 [2]	Art_0008 [2]	Art_0004 [2]	11 [2]	4.5 [2]	2.44 [2]	0 [D]	0 [D]
GP_2 [2]	Art_0008 [2]	Art_0005 [2]	11 [2]	4.5 [2]	2.44 [2]	0 [D]	0 [D]
GP_3 [2]	Art_0008 [2]	Art_0006 [2]	11 [2]	4.5 [2]	2.44 [2]	0 [D]	0 [D]
GP_4 [2]	Art_0004 [2]	Art_0009 [2]	4.5 [2]	20 [2]	4.44 [2]	0 [D]	0 [D]
GP_5 [2]	Art_0005 [2]	Art_0009 [2]	4.5 [2]	20 [2]	4.44 [2]	0 [D]	0 [D]
GP_6 [2]	Art_0006 [2]	Art_0009 [2]	4.5 [2]	20 [2]	4.44 [2]	0 [D]	0 [D]

Table 3.13: Artifact Associations for the Gearbox Derived from Assembly Constraints

ArtifactAssociation

Id	Name	Artifact_Id s	Assembly_ Constraints	Type
0001	FC7 [2]	0007, 0001 [2]	AC_1, AC_2 [3]	Conn
0002	FC8 [2]	0007, 0002 [2]	AC_3, AC_4 [3]	Conn
0003	FC9 [2]	0007, 0003 [2]	AC_5, AC_6 [3]	Conn
0004	MC2 [2]	0001, 0004 [2]	AC_7, AC_8 [3]	Conn
0005	MC3 [2]	0002, 0005 [2]	AC_9, AC_10 [3]	Conn
0006	MC4 [2]	0003, 0006 [2]	AC_11, AC_12 [3]	Conn
0007	PO1 [2]	0007, 0008 [2]	AC_13 [3]	PO
0008	PO2 [2]	0007, 0009 [2]	AC_14 [3]	PO

Table 3.14: Connections in the Gearbox Assembly

Connection

Id	Type	ParametricAssembly Constraint	Artifacts	AssemblyFeatures	KinematicPair
FC7 [2]	Fixed [2]	AC_1, AC_2 [3]	0007, 0001 [2]	PinHole3, PinCylinder3 [2,3]	Null
FC8 [2]	Fixed [2]	AC_3, AC_4 [3]	0007, 0002 [2]	PinHole4, PinCylinder4 [2,3]	Null
FC9 [2]	Fixed [2]	AC_5, AC_6 [3]	0007, 0003 [2]	PinHole5, PinCylinder5 [2,3]	Null
MC2 [2]	Movable [2]	AC_7, AC_8 [3]	0001, 0004 [2]	PinCylinder6, GearJournal_Surface_1 [2,3]	RP_2 [2]
MC3 [2]	Movable [2]	AC_9, AC_10 [3]	0002, 0005 [2]	PinCylinder7, GearJournal_Surface_2 [2,3]	RP_3 [2]
MC4 [2]	Movable [2]	AC_11, AC_12 [3]	0003, 0006 [2]	PinCylinder8, GearJournal_Surface_3 [2,3]	RP_4 [2]
MC5 [2]	Movable [2]	-	0004, 0008 [2]	Teeth_7, teeth_1 [2,4]	GP_1 [2]
MC6 [2]	Movable [2]	-	0005, 0008 [2]	Teeth_9, teeth_2 [2,4]	GP_2 [2]
MC7 [2]	Movable [2]	-	0006, 0008 [2]	Teeth_11, teeth_3 [2,4]	GP_3 [2]
MC9 [2]	Movable [2]	-	0004, 0009 [2]	Teeth_8, teeth_4 [2,4]	GP_4 [2]
MC10 [2]	Movable [2]	-	0005, 0009 [2]	Teeth_10, teeth_5 [2,4]	GP_5 [2]
MC11 [2]	Movable [2]	-	0006, 0009 [2]	Teeth_12, teeth_6 [2,4]	GP_6 [2]
MC12 [2]	Movable [2]	-	0010, Bearing [2]	Cylinder, journal [2]	RP_5 [2,3]

Table 3.15: Position/Orientations for the Gearbox Assembly

Position_ Orientation

Id	AssemblyConstraints	Artifacts	Mating Features
PO1 [2]	AC_13 [2,3]	0007, 0008 [2]	[2,4]
PO2 [2]	AC_14 [2,3]	0007, 0009 [2]	[2,4]

Since some features cannot be directly extracted from the parts' STEP files, a separate user interface is needed to input certain data into the OAM database. In some cases, we cannot define artifact associations at all by using 3D CAD modeling packages (e.g., gear teeth associations) in the assembly model. The AssemblyFeatures can be defined manually through the user interface (Table 3.16).

Table 3.16: Assembly Features, Defined Manually Using the User Interface

AssemblyFeature

Id	Name	Artifact	Definition / Parameters
0014	Teeth_1	0008	Teethform_1
0015	Teeth_2	0008	Teethform_1
0016	Teeth_3	0008	Teethform_1
0017	Teeth_4	0009	Teethform_1
0018	Teeth_5	0009	Teethform_1
0019	Teeth_6	0009	Teethform_1
0020	Teeth_7	0004	Teethform_1
0021	Teeth_8	0004	Teethform_1
0022	Teeth_9	0005	Teethform_1
0023	Teeth_11	0006	Teethform_1
0024	Teeth_10	0005	Teethform_1
0025	Teeth_12	0006	Teethform_1
0026	Cylinder	0010	L= width of bearing_Journal

The tolerance information generated for the sun gear has been shown in Figure 3.20, and the correct design data is tabulated in Tables 3.19 and 3.20.

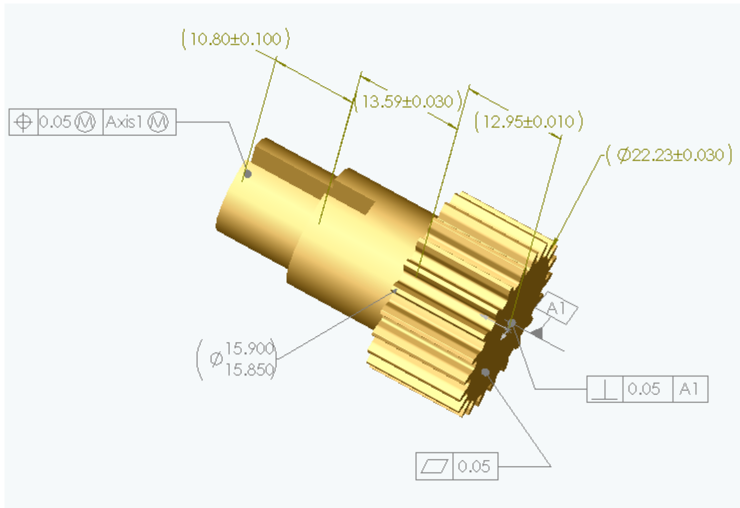


Figure 3.20: Geometrical and Dimensional Tolerancing on the SunGear (Art_0007)

In 3D modeling packages, parts are assembled using mating conditions. These mating conditions are associated with orientation and location tolerances. For the assembly representation, we need to connect these assembly constraints (mating conditions) and geometric (orientation and location) tolerances. When assembly conditions (e.g., concentric, parallel, etc.) are established, the related geometric tolerances (concentricity, parallelism) can be associated in the assembly/tolerance representation schema. As can be seen in Table 3.14, the "fc7" connection has three mating conditions, according to its assembly features. In regard to the Connection of this artifact association, the cylindrical surfaces are coaxial (concentric) and the planar surfaces are parallel. After creating an assembly representation scheme, we establish its tolerance representation as GeometricTolerances are listed in Table 3.17. For the connection "fc7," we describe relations between the assembly mate and its tolerance representation using the same connection Id. Then, appropriate relations between these "mates" and the geometric tolerances are established.

Table 3.17: Tolerance Class with Attributes

Tolerance

Id	Name	Type	Artifact	OAMF	Magnitude	Datum	MMC
GT_1 [3]	PerpTol_1 [3]	Or_Perp [3]	0008 [3]	OAMF_ EndSurface1[3]	0.05 [3]	DatumAxis_1 (A1) [3]	
GT_2 [3]	Flat_tol_1 [3]	Form_Flat [3]	0008 [3]	OAMF_ Endsurface1 [3]	0.05 [3]	-	
DT_1 [4]	Dim_Tol_1 [4]	Dim_Tol [4]	0008 [4]	OAMF_ Endsurface1 [4]	(ϕ22.23)±0.03 [4]		
DT_2 [4]	Dim_Tol_2 [4]	Dim_Tol [4]	0008 [4]	OAMF_SGear_teeth [4]	(12.95)±0.01 [4]		
GT_3 [3]	CylTol_1 [3]	Form_Cyld [3]	0008 [3]	OAMF_ Shank [3]	0.1 [3]		
DT_3 [4]	Dim_Tol_3 [4]	Dim_Tol [4]	0008 [4]	OAMF_ Shank [4]	(13.59)±0.03 [4]		
DT_4 [4]	Dim_Tol_4 [4]	Dim_Tol [4]	0008 [4]	OAMF_ Shank [4]	ϕ15.85~ ϕ15.85[4]		
GT_4 [3]	Pos_Tol_1[3]	Or_Pos [3]	0008 [3]	OAMF_inputshaft [3]	0.05 [3]	DatumAxis_1 (A1) [3]	MMC
DT_5 [4]	Dim_Tol_5 [4]	Dim_Tol [4]	0008 [4]	OAMF_inputshaft [4]	(10.85)±0.10 [4]		
GT_5 [3]	CylTol_2 [3]	Form_Cyld [3]	0004 [3]	PinHole6:AF [3]	0.02 [3]		
DT_6 [4]	Dim_Tol_6 [4]	Dim_Tol [4]	0004 [4]	PinHole6:AF [4]	(ϕ4.90)±0.01 [4]		
DT_7 [4]	Dim_Tol_7 [4]	Dim_Tol [4]	0004 [4]	GearCylinder1 [4]	(12.70)±0.10 [4]		
DT_8 [4]	Dim_Tol_8 [4]	Dim_Tol [4]	0004 [4]	GearCylinder1 [4]	(ϕ10.16)±0.01 [4]		
GT_6 [3]	Par_Tol_1 [3]	Form_Par [3]	0009 [3]	rimsurface [3]	0.05 [3]	DatumPlane_3 (A3) [3]	
DT_9 [4]	Dim_Tol_9 [4]	Dim_Tol [4]	0009 [4]	GearTeethHole [4]	(ϕ42.62)±0.01 [4]		
DT_10 [4]	Dim_Tol_10 [4]	Dim_Tol [4]	0009 [4]	PinHole1,2 [4]	(ϕ3.30)±0.05 [4]		
GT_7 [3]	PerpTol_2 [3]	Or_Perp [4]	0007 [3]	EndSurface2 [3]	0.03 [3]	DatumAxis_2 (A2) [3]	
GT_8 [3]	Flat_tol_2 [3]	Form_Flat [3]	0007 [3]	EndSurface2 [3]	0.06 [3]	-	
GT_9 [3]	TotRun_1 [3]	RO_Total [3]	0007 [3]	outputShaftShank [3]	0.1 [3]	DatumAxis_2 (A2) [3]	
GT_10 [3]	ProfSurfTol1[3]	Prof_Surf [3]	0007 [3]	Keyway [3]	0.1 [3]	-	

Chapter 4

Representation of Product Function and Behavior

4.1 Introduction

Product functionality is one of the important factors to be considered in product development. Although the definition of geometry is straightforward and well-represented in current information exchange standards (e.g., STEP, IGES, etc.), the current standards do not address how to represent the functions and behaviors of artifacts. Any knowledge of product functions helps users make intelligent decisions during product design. Though a function may be well known, it is handled at different levels of the product development lifecycle with different information content. The most common definition for function is what the artifact is intended to do. The behavior is the system's response to scenarios under a variety of conditions.

In product representation, the extraction of geometry has not been that difficult, but it is necessary to identify the semantic structures (i.e., features) for reasoning about a component's function. There are a number of methods for establishing function structure, as mentioned in Chapter 2.

In this chapter, a product information model is developed for representation of function, behavior, artifact and interrelations among them. Defining of interrelations, which mean that there cannot be any deficiency or disconnection among those entities, needs a common ground (i.e., a common parameter; e.g., degree of freedom), so, there will be a connection between them to check and verify the information of both side. Then, an important aspect of this work, traceability, is provided by interrelating them in a way that enables one to follow the functions of the object from its main functional requirements to its sub-functions, and to its structure and design rationale arguments.

4.2 Representation of Product Function

Functional information is handled throughout the lifecycle and includes the functional requirements (or purpose—for example, to transmit energy), the functional input and output, and the functional associations between an artifact and the environment (or between an artifact and other artifacts in other systems/assemblies, humans, or the environment itself). In conceptual design, the overall function is defined and decomposed into sub-functions, in preliminary design; artifact (a solution with shape and material information) comes into picture to perform the approximate functions. In detailed design, more functional features are introduced to the system to fulfill the more precise functions. So, the relationship between function and artifact is specified in more detail with intended behavior information.

Regardless of variations in methodology, all functional modeling begins by formulating the overall product function. When the overall function of the product is broken into small, easily solved sub-functions, the form of the product follows from the assembly of all sub-function solutions. The input/output flows (Figure 4.1) are most easily established after the development of a set of customer needs for the product. Many researchers have adapted and extended this input-output perspective [Gero, 1990; Stone & Wood, 1999, 2000; Gorti & Sriram, 1996; Szykman et al., 1999; Otto & Wood, 2001; Kirschman & Fadel, 1998]. Figure 4.2 shows an example for the "Reduce Speed" function and input/output flow.

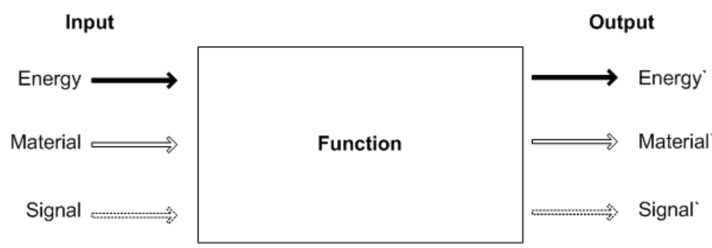


Figure 4.1: A Block Representation of a Function

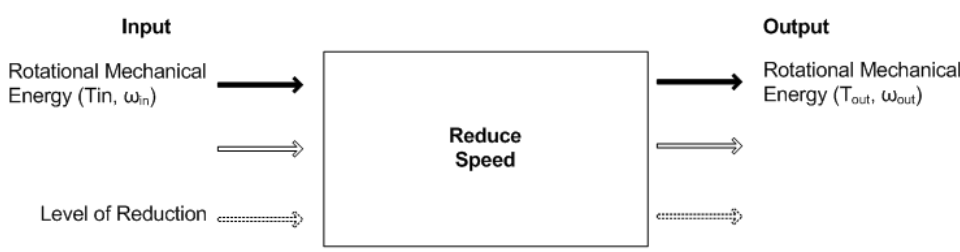


Figure 4.2:A Block Representation of "Reduce Speed" Function

4.2.1 Functional Decomposition

Initially, functional requirements are used to define the main function. This overall function has to be divided into identifiable sub-functions; consequently, artifacts can be assigned for those lower-level functions. To eliminate getting confused by using different words for similar actions, NIST (Hirtz, 2002) brought the most common two function taxonomies together and came up with the function set for standardization of function terms used in literature. Function set is shown in Table 4.1 and descriptions can be found in Appendix -2. By the arrangement and grouping of individual sub-functions, a function structure is developed for the overall function. A function structure breakdown with flow, adapted from Pahl and Beitz (2007), is shown in Figure 4.3.

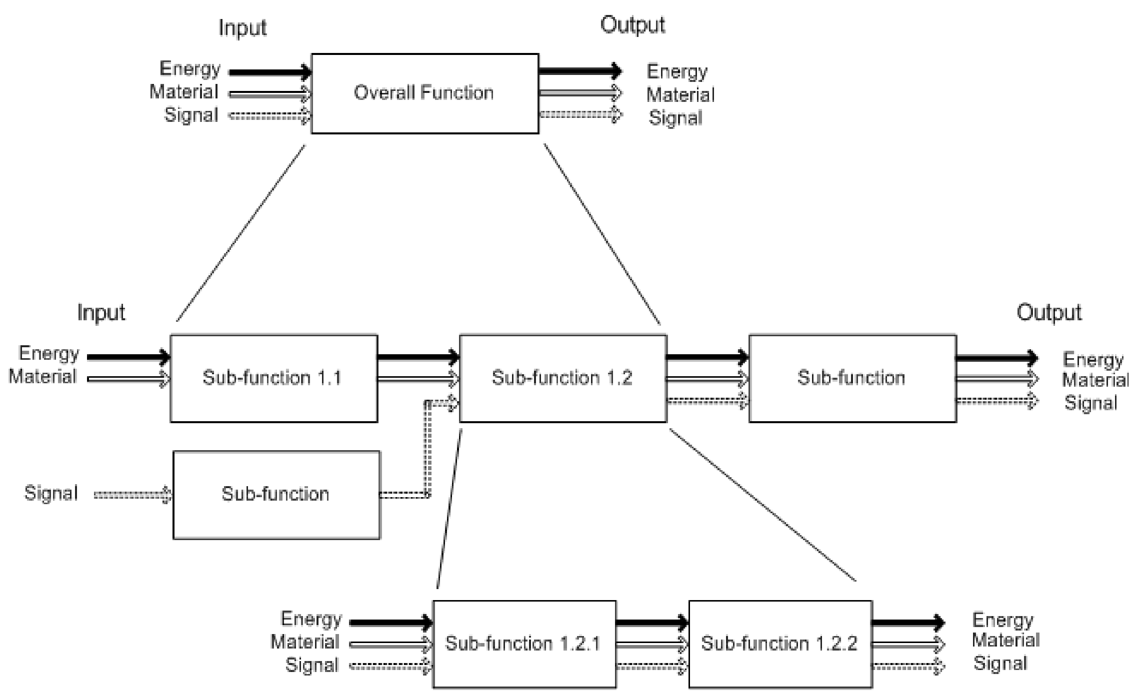


Figure 4.3: Function structure breakdown with flow (adapted from Pahl & Beitz, 2007)

Table 4.1: Functional Basis Reconciled Function Set (Hirtz et al., 2002)

Class (Primary)	Secondary	Tertiary	Correspondents
Branch	Separate		Isolate, sever, disjoin
		Divide	Detach, isolate, release, sort, split, disconnect, subtract
		Extract	Refine, filter, purify, percolate, strain, clear
		Remove	Cut, drill, lathe, polish, sand
	Distribute		Diffuse, dispel, disperse, dissipate, diverge, scatter
Channel	Import		Form entrance, allow, input, capture
	Export		Dispose, eject, emit, empty, remove, destroy, eliminate
	Transfer		Carry, deliver
		Transport	Advance, lift, move
		Transmit	Conduct, convey
	Guide		Direct, shift, steer, straighten, switch
		Translate	Move, relocate
		Rotate	Spin, turn
		Allow DOF	Constrain, unfasten, unlock
Connect	Couple		Associate, connect
		Join	Assemble, fasten
		Link	Attach
	Mix		Add, blend, coalesce, combine, pack
Control	Actuate		Enable, initiate, start, turn-on
	Regulate		Control, equalize, limit, maintain
		Increase	Allow, open
		Decrease	Close, delay, interrupt
	Change		Adjust, modulate, clear, demodulate, invert, normalize, rectify, reset,
		Increment	Amplify, enhance, magnify, multiply
		Decrement	Attenuate, dampen, reduce
		Shape	Compact, compress, crush, pierce, deform, form
		Condition	Prepare, adapt, treat
	Stop		End, halt, pause, interrupt, restrain
		Prevent	Disable, turn-off
		Inhibit	Shield, insulate, protect, resist
Convert	Convert		Condense, create, decode, differentiate, digitize, encode, evaporate,
Provision	Store		Accumulate
		Contain	Capture, enclose
		Collect	Absorb, consume, fill, reserve
	Supply		Provide, replenish, retrieve
Signal	Sense		Feel, determine
		Detect	Discern, perceive, recognize
		Measure	Identify, locate
	Indicate		Announce, show, denote, record, register
		Track	Mark, time
		Display	Emit, expose, select
	Process		Compare, calculate, check
Support	Stabilize		Steady
	Secure		Constrain, hold, place, fix
	Position		Align, locate, orient

Thus, a function structure will have appropriate sub-functions. The individual sub-functions are simpler than the overall function and, furthermore, one can see which sub-function provides the most suitable starting point for matching appropriate artifacts. In the "Reduce Speed" function, "Receive Energy" is an important sub-function that will help us find the appropriate solution (artifact) with the working principle upon which the others clearly depend (see Figure 4.4). First, we should start from this sub-function (Receive Energy).

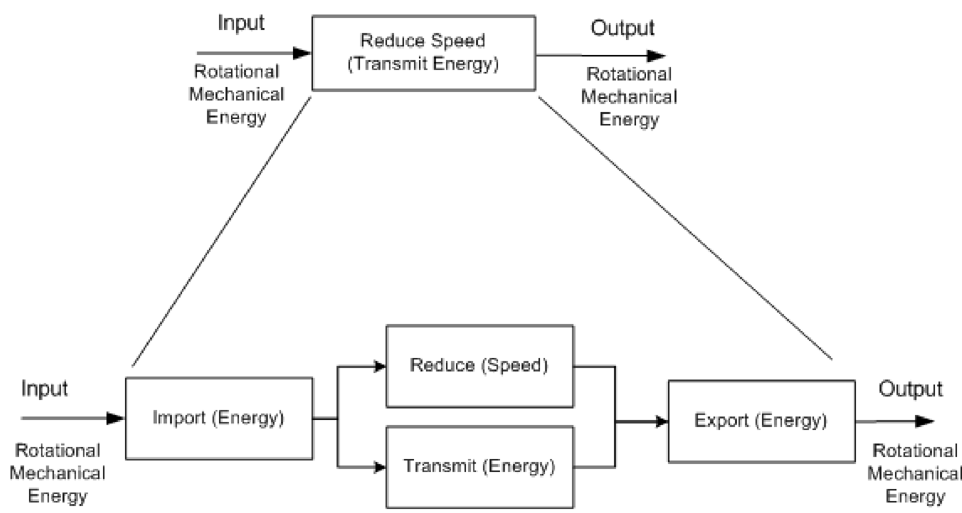


Figure 4.4: Functional Decomposition of the "Reduce Speed" Function

To fulfill all these functions, appropriate artifacts are selected from the alternatives, so functions are connected with artifacts for the first time, but without detailed information. After a basic function structure with connections has been formed, it will be easier to move on to the next step. For this phase, creating a temporary product structure for the basic function structure will be very useful. By doing that, we will be able to define more detailed sub-functions, auxiliary functions and the connections among all the functions. For example, for the “Reduce Speed” function, a planetary gear set (for its structure see Figure 4.5) is temporarily selected to allow the further detailing of functional decomposition. With the product structure information, it is easier to think about the lower-level sub-functions needed to fulfill upper-level functions. Subsequently, individual lower-level functions are assigned for parts and sub-assemblies in the planetary gear set structure, shown in Figure 4.6.

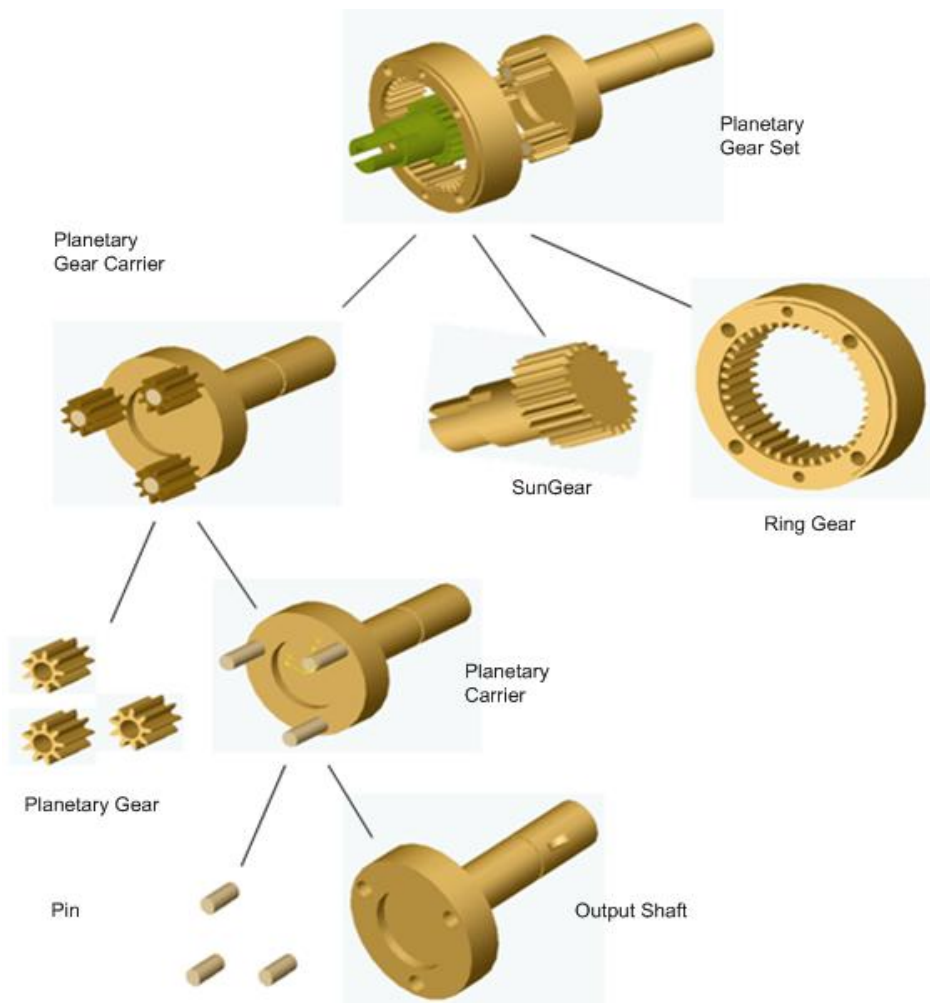


Figure 4.5: Assembly Structure of a Planetary Gear Set

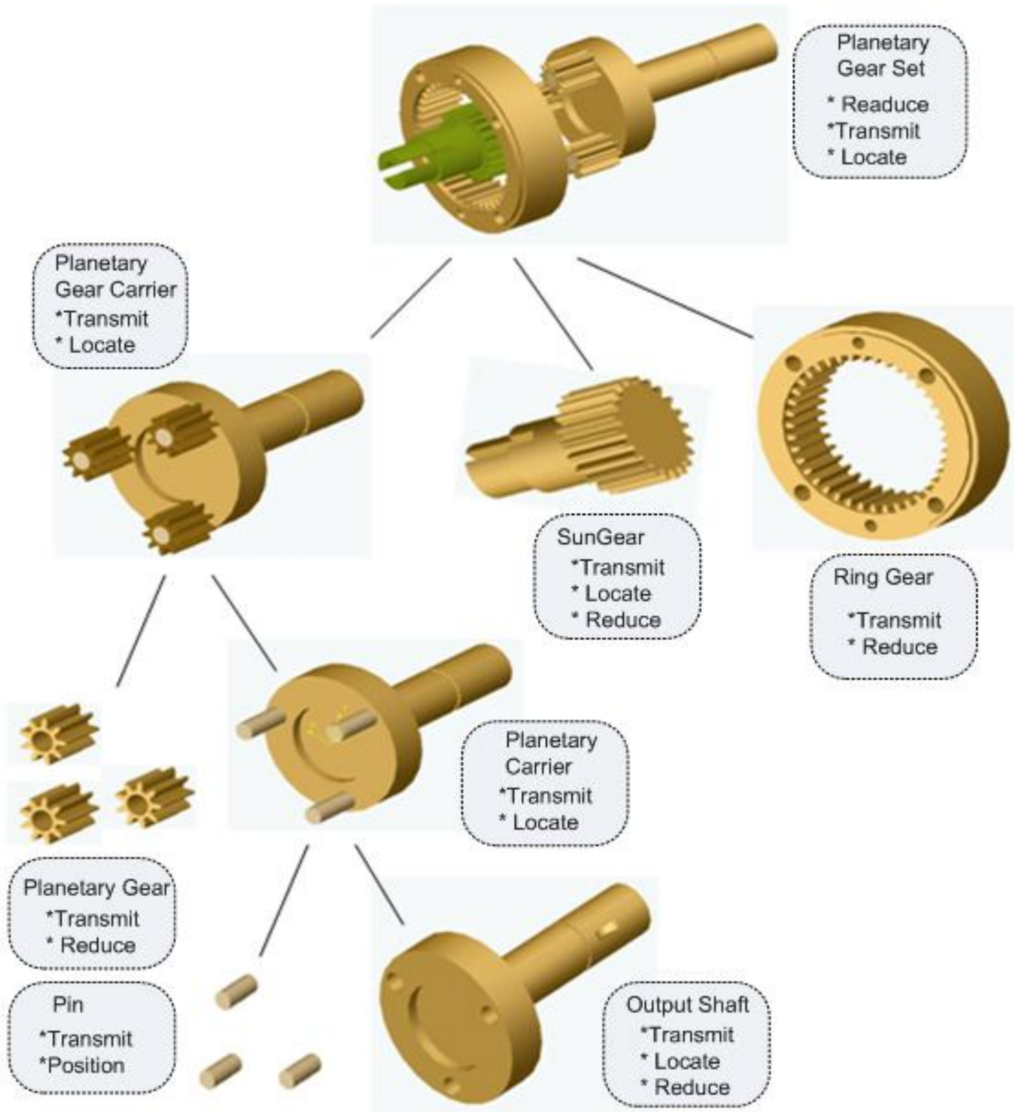


Figure 4.6: Artifacts and their Functions for the Required Functions

4.2.2 Flow

Flow is defined by Stone (2000) as "the representation of the quantities (entities) that are input and output by functions." The CPM model defines Flow as a medium (energy, material, message stream, etc.) that serves as the output of one or more transfer function(s) and the input of one or more other transfer function(s). In the literature, there are many flow definitions, and many components have been considered. As in Table 4.1 NIST (Hirtz, 2002) also came up with the reconciled flow set given in Table 4.2.

Table 4.2: Functional Basis Reconciled Flow Set (Hirtz et al., 2002)

Class (Primary)	Secondary	Tertiary	Correspondents
Material	Human		Hand, foot, head
	Gas		Homogeneous
	Liquid		Incompressible, compressible, homogeneous,
	Solid	Object	Rigid-body, elastic-body, widget
		Particulate	
		Composite	
	Plasma		
	Mixture	Gas-gas	
		Solid-solid	Aggregate
		…	
		Colloidal	Aerosol
Signal	Status	Auditory	Tone, word
		Olfactory	
		Tactile	Temperature, pressure, roughness
		Taste	
		Visual	Position, displacement
	Control	Analog	Oscillatory
		Discrete	Binary
Energy	Human		
	Acoustic		
	Biological		
	Chemical		
	Electrical		
	Electromagnetic	Optical, Solar	
	Hydraulic		
	Magnetic		
	Mechanical	Rotational, Translational	
	Pneumatic		
	Radioactive/Nuclear		
	Thermal		

Finally, the function structure, including all lower-level sub-functions, and the input-output flows between them, is developed in sequence to fulfill the overall “Reduce Speed” function; this is shown in Figure 4.7.

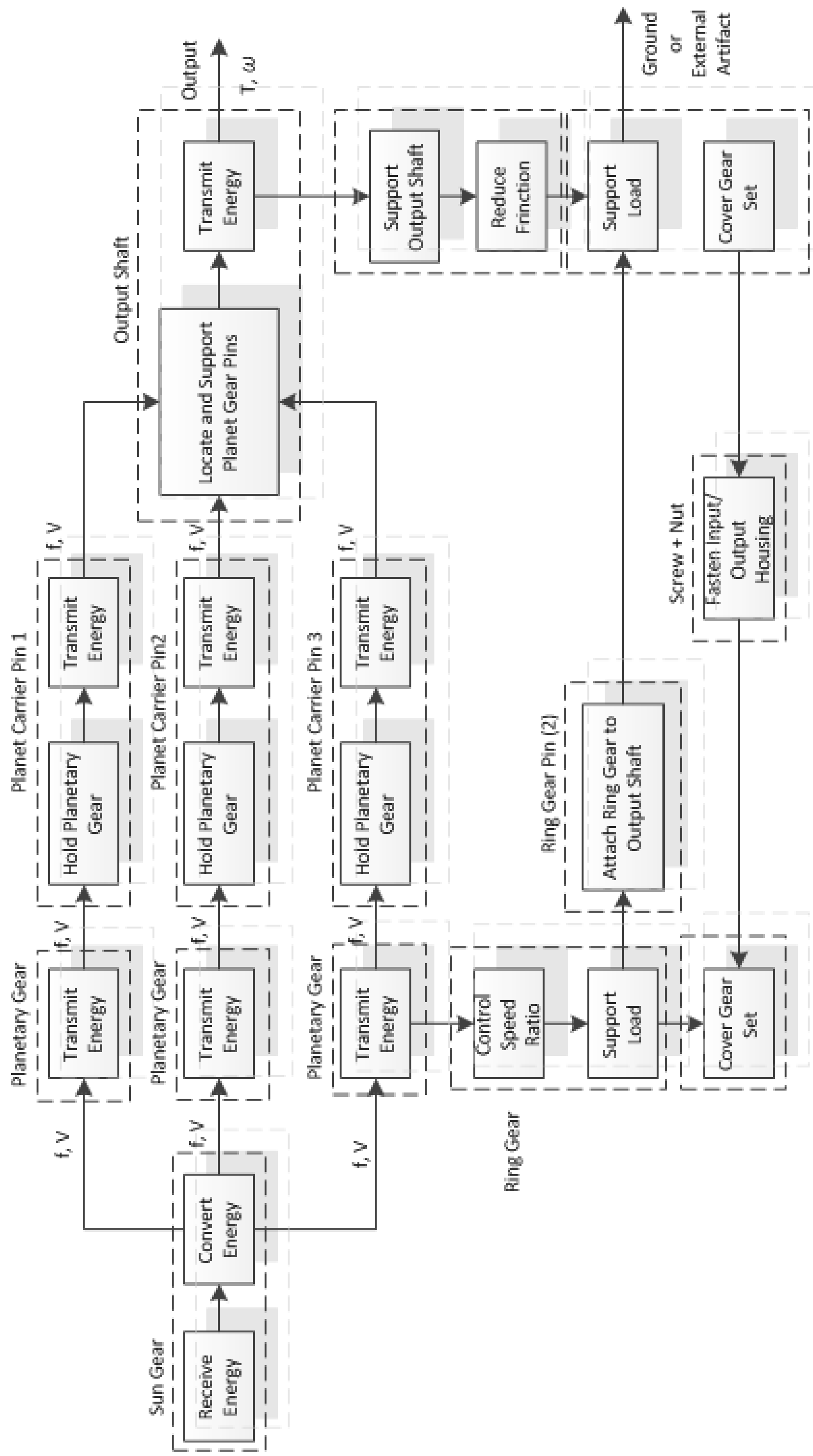


Figure 4.7: Planetary Gear Set and the Functions that Fulfill the Overall Function

Lower-level energy flow sets (i.e., translational mechanical energy) have power variables (i.e., force and velocity) to define the flow between entities (Table 4.3).

Table 4.3: Power Variables for the Energy Class of Flows (Hirtz et al., 2002)

Energy	Power Variables	
Systems	Effort (e) Analogy	Flow (f) Analogy
Human	Force (F)	Velocity (v)
Acoustic	Pressure (P)	Particle Velocity (v)
Biological	Pressure	Volumetric Flow
Mechanical -Translational	Force (F)	Velocity (v)
Mechanical - Rotational	Torque (t)	Angular Velocity (w)
Electrical	Voltage (V)	Current (i)
Electromagnetic - Optical	Intensity	Velocity (v)
Electromagnetic - Solar	Intensity	Velocity (v)
Hydraulic	Pressure (P)	Volume flow rate (dQ/dt)
Pneumatic	Pressure (P)	Mass Flow rate (dm/dt)
Thermal	Temperature (T)	Entropy change rate (ds/dt)
Thermal	Pressure (P)	Volume change rate (dV/dt)
Radioactive/Nuclear	Intensity	Decay Rate
Chemical	Affinity	Reaction rate
Chemical	Chemical potential (m)	Mole flow rate (dN/dt)
Chemical	Enthalpy (h)	Mass flow rate (dm/dt)
Magnetic	Magneto-motive force (em)	Magnetic flux rate (f)

4.2.3 The Representation of Function and Flow in the Modified CPM

In the Core Product Model (CPM), Function represents one aspect of what the artifact is supposed to do. It is also often used synonymously with the term *intended behavior* in literature. The Function class has information about function types, flows and functional parameters. A TransferFunction is a specialized form of Function involving the transfer of an input flow into an output flow. Examples of transfer functions are “Transmit” (a flow of fluid or current, or a message, etc.) and "convert" (from one energy flow to another or from a message to an action).

In addition to TransferFunction, there are two more function types added to the CORE model: StoreFunction and SupplyFunction. StoreFunction has input flow but no output flow, while SupplyFunction has output flow but no input flow. The Function class in the modified CPM also has attributes for function structure information, like sub-functions, as well as order information that specifies which function has to be fulfilled before another (see Figure 4.8).

Function types prescribe many restrictions to flows. For instance, the function "Decrement" must include input and output flows of the same type, whereas the function "Convert" needs to have distinct input and output flows. The relationship between some function types and their input-output flows is given in Table 4.4, which shows whether they must have the same or a different type of input-output flow, or whether they must have only input or only output flow.

Table 4.4: The Relationship between Function Type and Input-Output Flows

Function Type	InputFlow ...... OutputFlow
Transmit	= (Same type)
Change	= (Same type)
Regulate	= (Same type)
Convert	≠ (Different type)
Store	No OutputFlow
Supply	No InputFlow

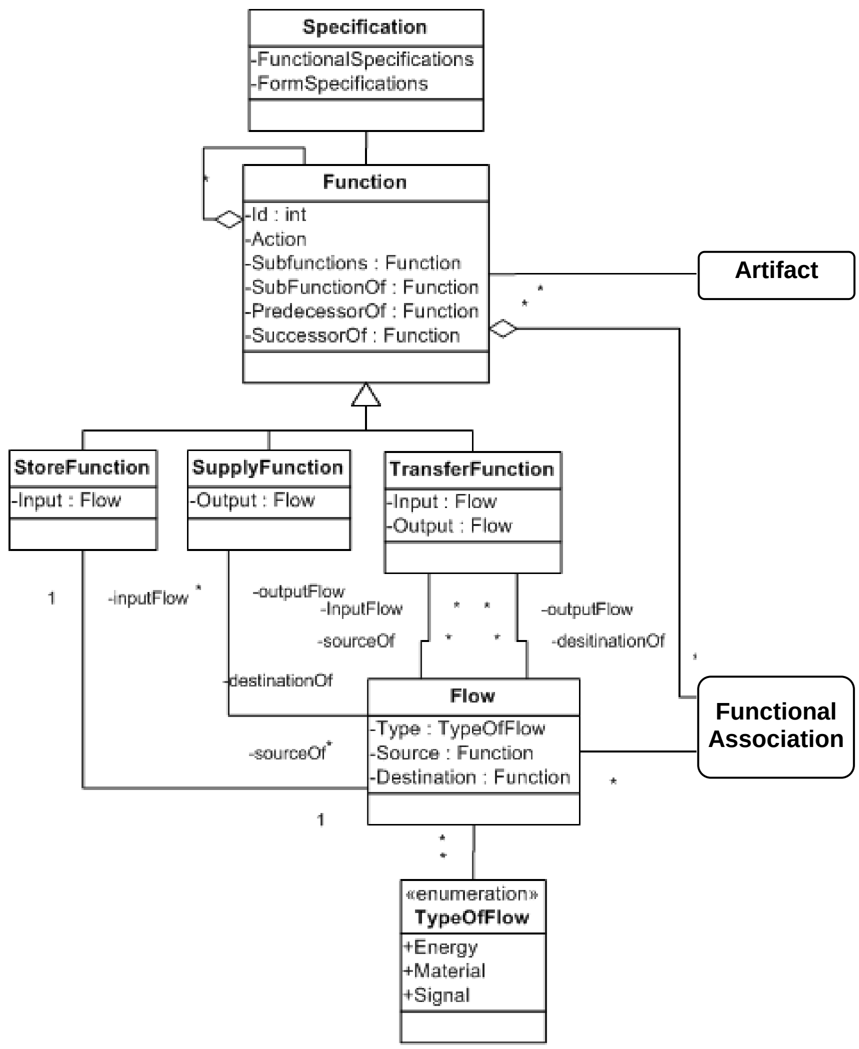


Figure 4.8:Representation of Function and Flow Classes in the Modified CPM

4.3 Functional Associations

There are many associations between two parts, in terms of structure, function, behavior, tolerance, kinematics, etc. These associations need to be represented in a consistent way, so that they will not conflict with each other. Functional associations are defined to serve this purpose. Functional associations are defined based on functional and form requirements and couple the related artifacts based on the spatial relationships among them. For example, an artifact which has a “Support Load” function is connected to another artifact with a “Transmitting Energy” functional association through their relevant assembly features, which have assembly feature association. Functional associations are also defined based on design requirements. For example, when a certain clearance between two surfaces has to be maintained between two artifacts, a functional association such as “Maintaining Clearance” is defined.

Functional associations are also used to trace the required functional and behavioral information in the assembly structure. If position, orientation, joint type, etc., in the assembly structure are changed for any reason, naturally, the function and behavior will be affected accordingly. Functional associations define the links to provide a consistent product information model, and also are used to trace all associated and affected artifacts. Then, the function and behavior model is modified to consider the new information, since all inputs and outputs are defined in terms of the associations. In this section, the function and its relations to the assembly structure are described. For example, in the planetary gear set, all the associated and required artifacts for the “Reduce Speed” function (ω_o / ω_i) provided by the model is given in Figure 4.9. If any of those artifacts is missing or there, is a disconnection (in terms of input/output flow, “Reduce Speed” function will not function properly or not function at all. Similarly, for the “Transmit (Energy)” function, all associated artifacts are shown in Figure 4.10.

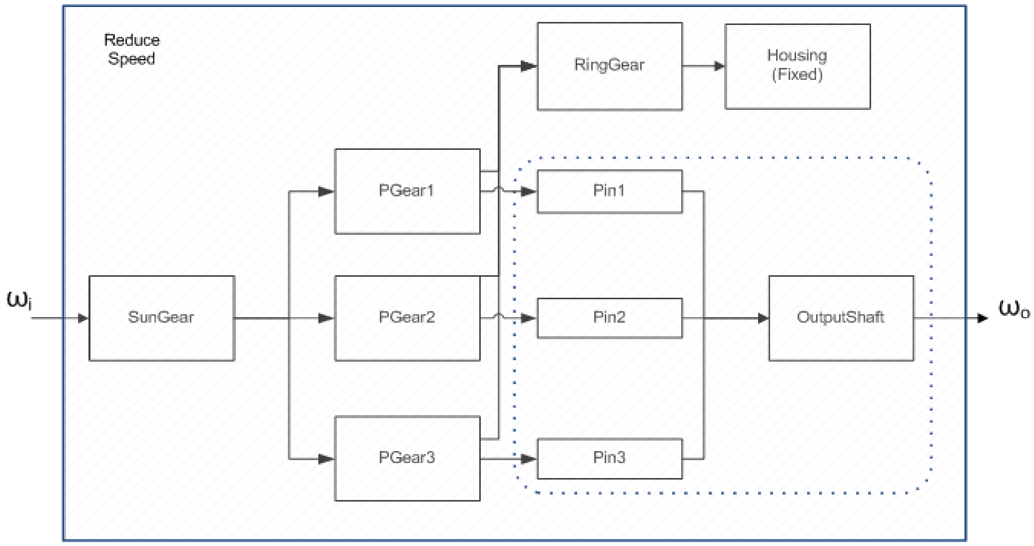


Figure 4.9: Functionally Associated Artifacts for the “Reduce Speed” Function

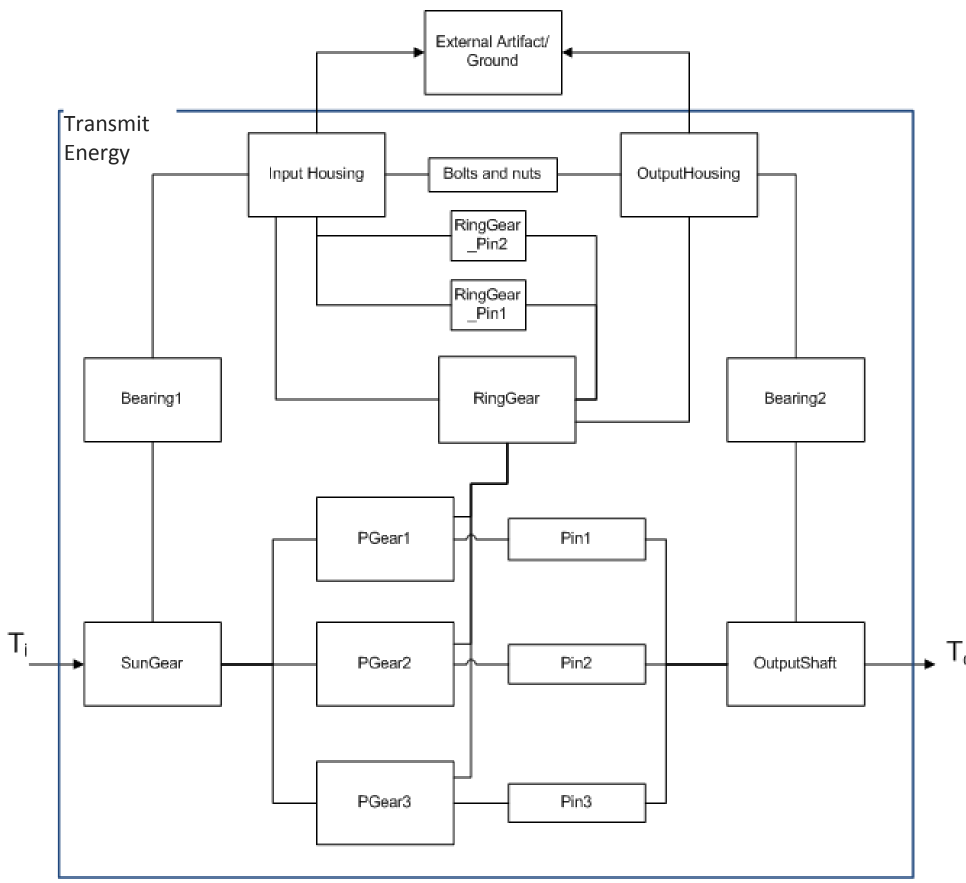


Figure 4.10: Functionally Associated Artifacts for the “Transmit (Energy)” Function

The partial UML class diagram in Figure 4.11 shows how Function and Flow are connected to Artifact and its Ports through the FunctionalAssociation class. Each artifact has one or more functions with flow (i.e., energy, material or signal) information. Artifacts are functionally connected through functional associations through their specialized features (ports). FunctionalAssociation also connects artifact and it's Function to its Behavior.

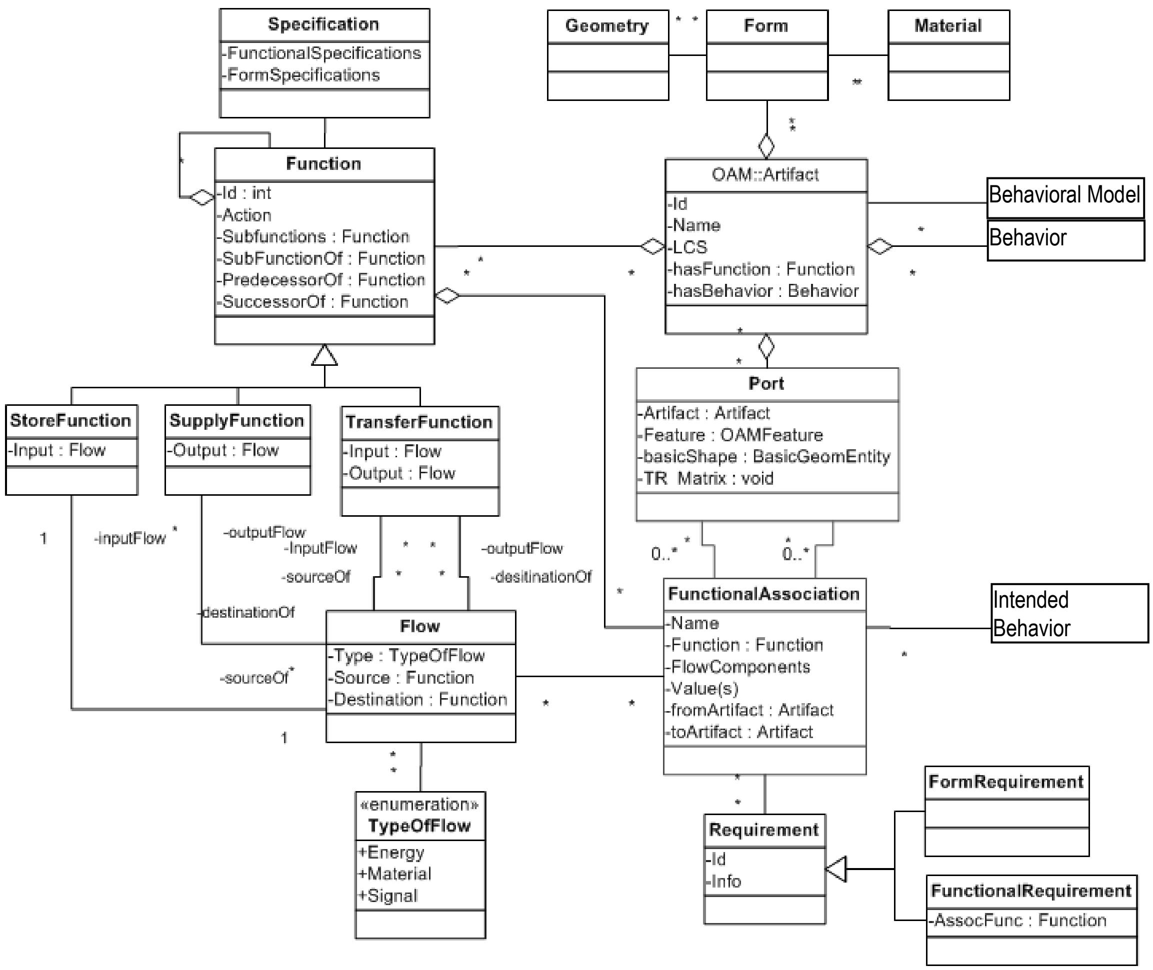


Figure 4.11: Representation of the FunctionalAssociation Class

As it is mentioned earlier, functional associations are defined among an artifacts' spatial and design relations (e.g., maintaining clearance between two surfaces). For a gear box, the main functions are "Transmit Energy" and "Reduce Speed". For the "Transmit Energy" function, the relationship among associated artifacts/features can be defined by the mechanical energy couples (i.e., f-v: force – velocity and M – ω: moment- angular velocity) as defined in Table 4.3. The functions and the functional input/outputs through ports of the planetary gearbox artifacts, such as SGN1: port of SunGear for input rotational mechanical energy) are shown in Figure 4.12.

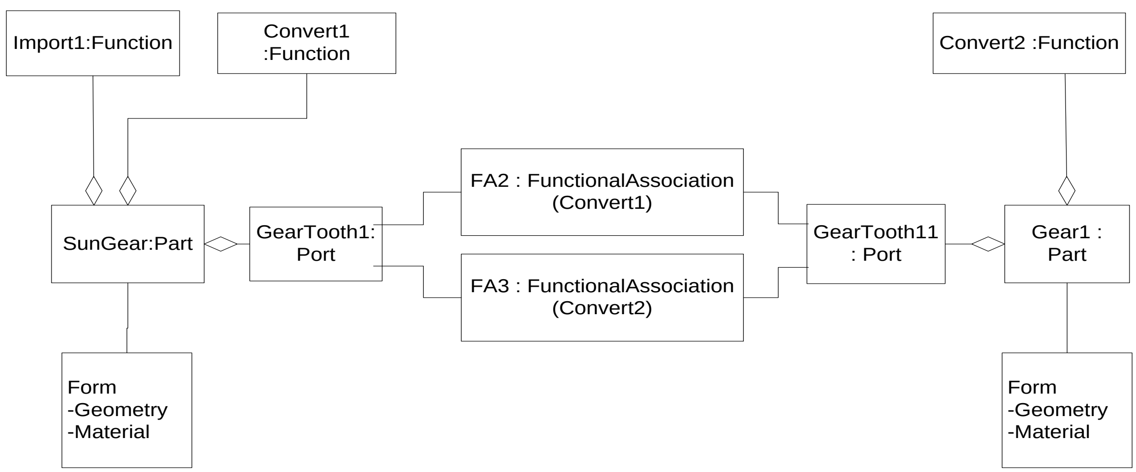


Figure 4.12: Functional Associations and Associated Classes in the modified CPM

Next few figures show associations in the Planetary Gearbox in different abstract levels. Figure 4.13 shows artifacts and functional and artifact associations among artifacts through their ports. Figure 4.14 shows functions and associated ports of SunGear and PlanetGear1. Figure 4.15 and Figure 4.16 give details of Artifacts, Ports with basic shape and LCS/FCS, and AssemblyFeatureAssociation including details of Gear Pair.

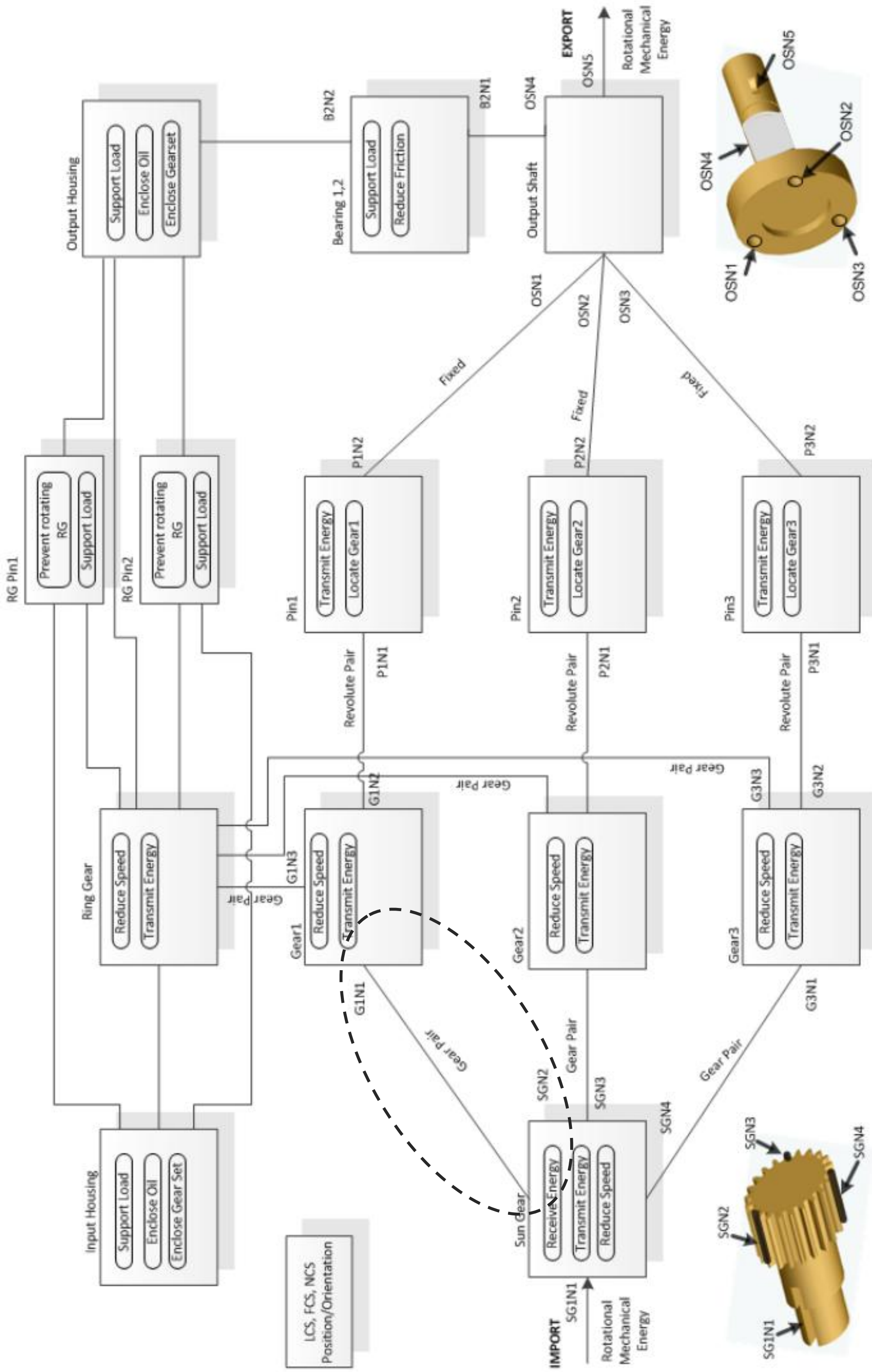


Figure 4.13: Functional Associations in the Planetary Gear Set

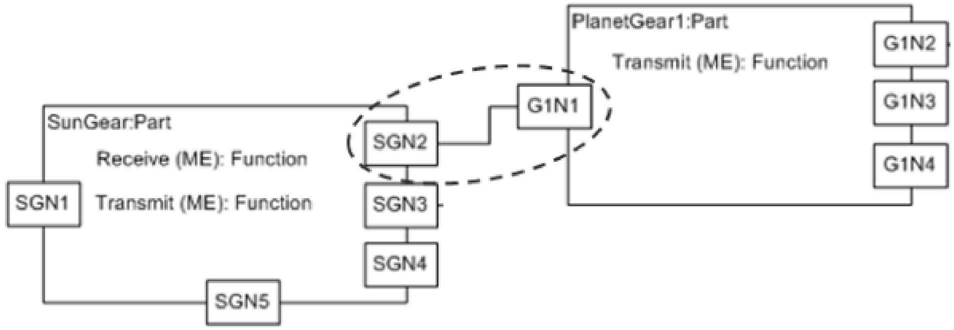


Figure 4.14: Functional Association between SunGear and PlanetGear1

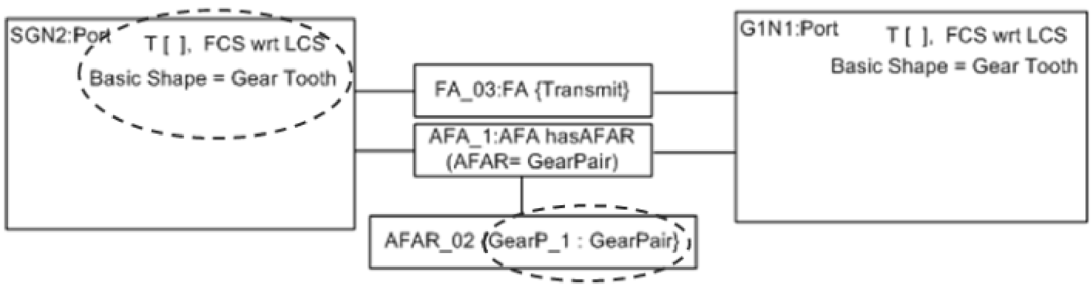


Figure 4.15: Feature Level Associations between SGN2 Port and G1N1 port

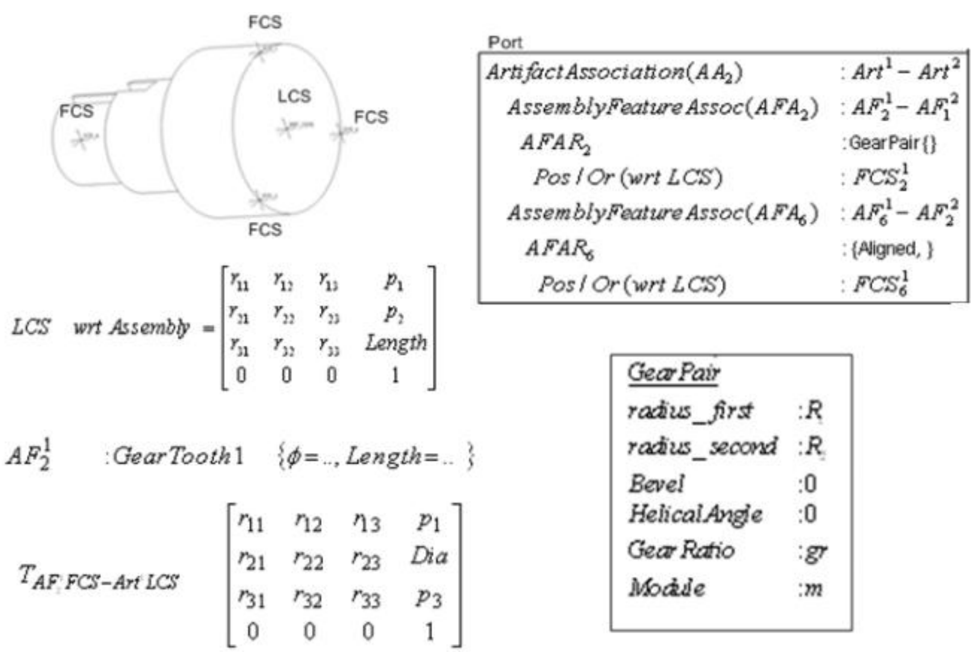


Figure 4.16: Details of Artifact, Port and AssemblyFeatureAssociation including KinematicPair

4.4 Representation of Behavior

Behavior is commonly defined as "the response of something to its environment." In our model, we define an artifact's behavior as a result of the interactions of the artifact with other artifacts in the assembly and with the environment, through a set of relevant functional associations. In the CPM, the behavior describes how the artifact implements its function. Behavior in the CPM is an abstract class and is specialized according to (1) intended behavior, (2) estimated (designed) behavior, (3) observed behavior, (4) unintended behavior, and (5) evaluated behavior. IntendedBehavior is defined as how it is supposed to fulfill the function based on customer needs and design specifications. On the other hand, the EstimatedBehavior (designed behavior) is the defined behavior in design, and considers only identified functional inputs and known (conceivable) environmental effects. The connection between function and behavior consists of the functional associations among features/artifacts.

Behavior is governed by engineering principles that are incorporated into a behavioral or causal model. Considering these functional inputs and functional associations, the BehaviorModel is developed based on relevant physical laws and engineering formulas. Application of the behavioral model to the artifact describes or simulates the artifact's EstimatedBehavior based on its form. For example, the planetary gear has three functional associations among the sun gear, ring gear and pin, according to artifact associations. The parameters of force, moment, velocity, etc., come through these functional associations, which affect the behavior of the gear. The planetary gear can then be designed with engineering formulas and Form (material and geometry) information, and the behavioral model will simulate the estimated behavior of the planetary gear assembly. The ObservedBehavior is defined as the artifact's actual behavior in its environment. These observed behaviors are then evaluated based on the requirements and engineering specifications defined in the IntendedBehavior, if the whereby the results of this evaluation process give the EvaluatedBehavior whether the result is accepted or not. It also includes key factors used for the decision. Unintended behavior can be a result of the evaluation process and of observed behaviors, like heat generation in the gear box. Since our model allows for it, new functions can be added for the unintended behaviors.Figure 4.17 shows the interrelationships among the function, behavior and form of an artifact.

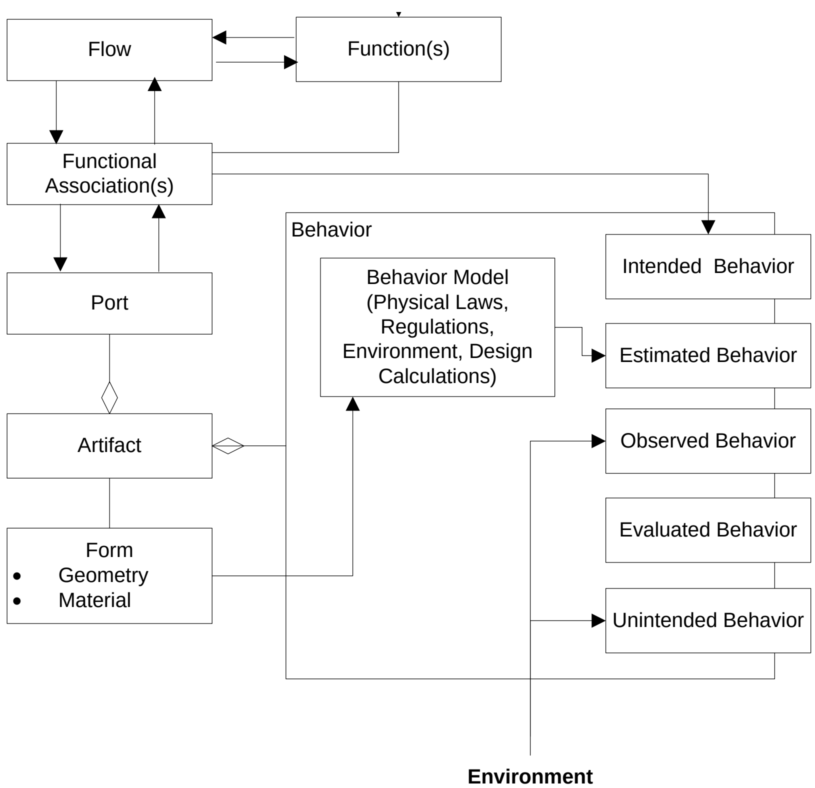


Figure 4.17: Representation of Behavior, Function and Artifact

An artifact has a function or functions, and these functions have inputs and outputs in terms of energy, material and signal, with parameter, position/orientation and associated artifact information. In addition to that, a function also has sequence information (what function comes before/after what function) and functional association(s). Functional associations between artifact function and behavior are introduced to the CPM model in Figure 4.18, and the entire modified CPM is shown in Figure 4.19.

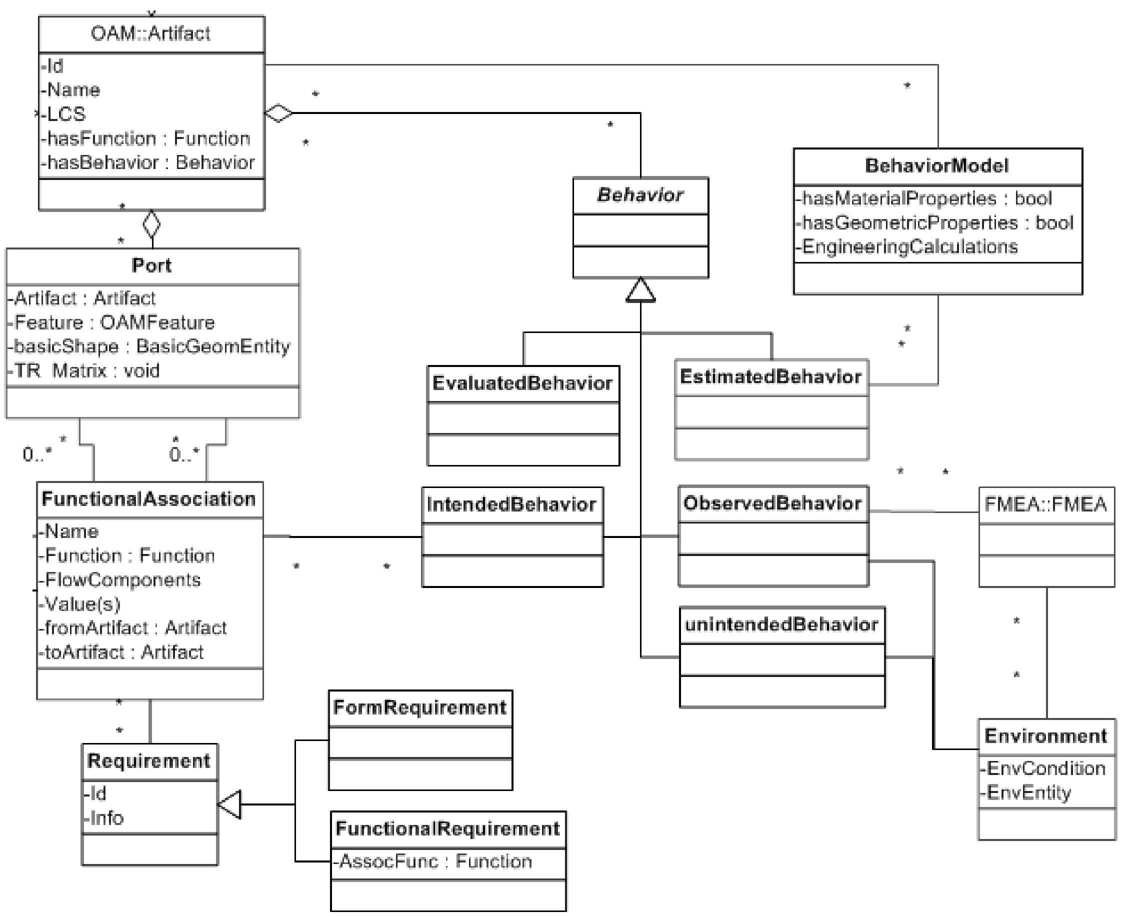


Figure 4.18: Relationships among FunctionalAssociation, Behavior and Form in the modified CPM

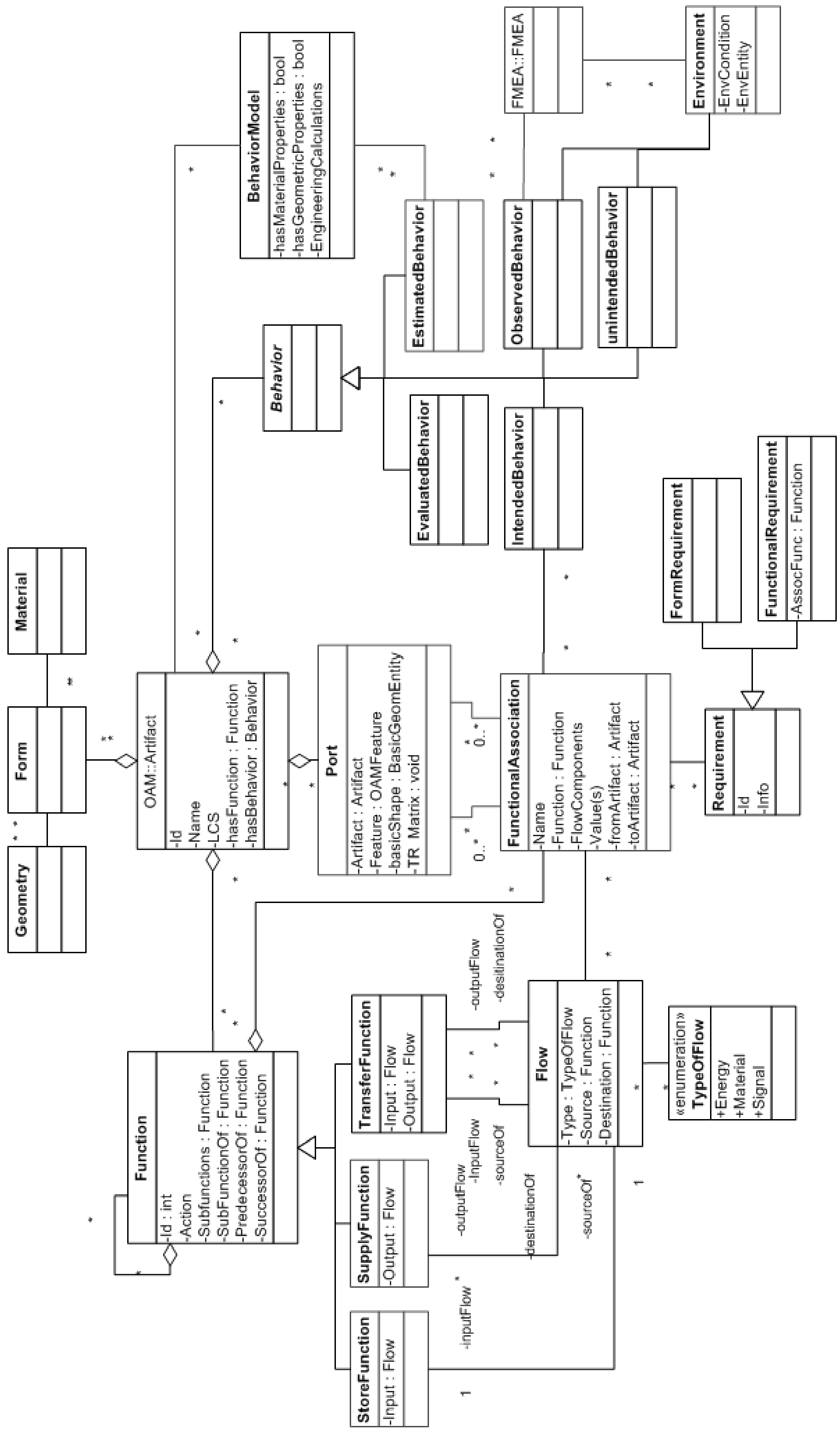


Figure 4.19: Associations among Function, Flow and Artifacts

Based on the behavior of the artifact, unintended behavior (e.g., heat generation) might result and new functions (e.g., the removal of heat) might have to be introduced to the model to overcome unintended behaviors. As a result, the function behavior model needs to be updated and to be allowed to add the new functions based on the behavior of the artifact or new requirements. At the same time, the evaluation process is stored in the model, so that anyone can see the process and use the information. For Pin1: Artifact, the functions, functional associations and behavior (with the behavioral model for estimated behavior) and the interrelationships among them are shown in Figure 4.20.

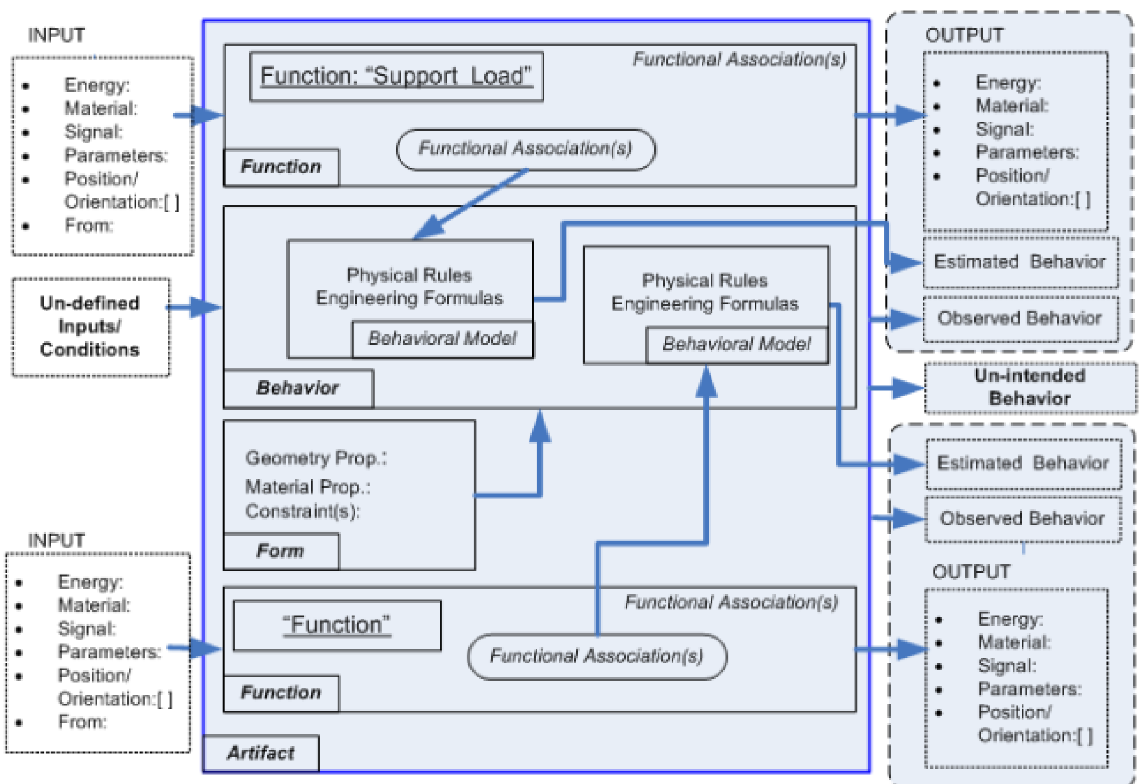


Figure 4.20: Interrelationships among the Function, Behavior and Form of the Artifact

4.4.1 Failure Mode and Effects Analysis in Behavior Model

In Bertsche (2008), Failure Mode and Effects Analysis (FMEA) is defined as a "systematical approach which discovers all modes of failure for arbitrary systems, sub-systems and components". At the same time, likely failure causes and effects are introduced. The procedure ends with risk evaluation and requirements for optimization actions. This method purports to identify at an early stage the risks and weak points of a product, so as to allow for timely improvements in execution. More precisely, IEEE Std 352-1975: "Guide for general principles of reliability analysis of nuclear power generating station protection systems," defines the major objectives of an FMEA.

In this study, the FMEA technique is discussed for defining the behavior of an artifact, based on predictions and possible failure modes and their effects. As this will help in tracing the problems in the product, it will also help the designer to define evaluation criteria for artifact selection. In this way, the designer will have a parameter to check how environmental or any other changes in conditions might affect the behavior of the artifact. Table 4.5 gives an example of FMEA for a gear box. For example, as shown in Figure 4.5, for the function "Seal", it will provide information about possible failures and causes, potential effects, and recommended actions to overcome these problems.

Next step will be detailing the connection between FMEA and behavior in CPM for better understanding of observed and un-intended behaviors of an artifact.

Table 4.5: Failure Mode Effect Analysis (FMEA) for Gear Boxes

Artifacts	Function	Failure Mode	Cause(s)	Effects	Indirect Effects	Recommended Actions
Gear	Reduce Speed Transfer Energy	Wear Failure Polishing Moderate Wear Extreme Wear Abrasive Wear Corrosive Wear	Lubricant film is too thin Lubricant temperature Inappropriate lubricant Metal particles, dirt, rust etc. in lubricant Lubricant is contaminated with acid, water etc.	Tooth profile destroyed Gear tooth wear Tooth surface deformation	Bearing Damage	Higher viscosity lubricant Lower lubricant temperature Use appropriate lubricant Lubricant filtering system
		Surface Fatigue Fail. Pitting Failures Destructive Pitting Spalling Micro pitting Case Crushing	Surface contact stress Number of stress cycles Overloading High surface load and high temperature Heavy loading (on case hardened gears)			Manufacturing control of involute profile Reduce loading below endurance limit Redesign (increase tooth width) if possible Improve surface finish through honing, grinding
		Plastic Flow Failure Rippling Ridging	High contact stress combined with rolling and sliding High contact stress combined with rolling and knead High contact stress and low sliding velocity		Loss of power	Reduce the contact surface increase the hardness Proper heat treatment
		Breakage Failure Bending Failure Brkg Overload Breakage Random Fracture	Excessive tooth load Stress risers Bearing seizure Foreign material between gear mesh			
Bearing	Support rotating load	Bearing Failure	Insufficient commissioning of the bearing set Use of improper and non-compatible oil seals use of wrong and no approved lubricants	Misalignment		Use appropriate sealing and lubricant
Sealing	Seal to retain oil Exclude contami nants	Fail to seal	Corrosion of the running surface of the shaft	Bearing Damage	Gear tooth damage	Replace oil Replace sealing element

4.5 Mathematical Definitions for Establishing Traceability

It is necessary to describe the object (an instance of function, behavior, feature, etc.) interrelations in such a way that they enable one to follow the object through the associations in the assembly (e.g., function: from main functional requirements to sub-functions). There cannot be any deficiency in, or disconnection among, these entities. In other words, the information model has to be complete in terms of the traceability of function, behavior, and spatial associations, in order to support all information exchange activities. To enable the formation of a consistent, complete and traceable information model, all the associations are defined in a mathematical way. For this reason, the positions and orientations of the parts, the features (in basic shape level) of these parts and other information nodes, are first defined with transformation matrices similar to the Constructive Solid Geometry (CSG) representation. Second, the connections (joints) between parts and between features are defined, based on degrees of freedom, by twist matrices. The wrench matrices are defined based on the twist matrices, to show the resultant forces and moments acting on rigid bodies. Third, a behavior model, which includes related physical rules and functional associations based on engineering formulas, is defined manually with artifact's form information. All of these matrix-based definitions—for positions/orientations, connections, functional input/outputs—and the behavioral model are then utilized to develop a mathematical model. When many parts are joined this way, one can navigate from part to part by following the transformation frames.

4.5.1 Screw Theory-Based Representations of Assembly Associations

A screw is a method of demonstrating the motions of a rigid body or the forces and moments acting on it. In this work, it is used to link assembly associations (i.e., motion constraints and d-o-f) to artifact's behavior. It is clear that when any change happened in an assembly association between two artifacts, like loose of a constraint in a structure, the relevant reactant forces and/or moments will accordingly be changed and artifact might behave in a different way. Therefore, the necessary link between assembly constraints and behavior is provided by application of "Screw Theory". Screws that characterize motions are called *twists* or *twist matrices (TR)*, whereas screws that characterize forces are called *wrenches* or *wrench matrices (WR)*. A twist or wrench matrix consists of six columns and one to six rows, one for each degree of freedom. These matrices can depict a host of part-to-part constraints, and they are used to build a toolkit of useful assembly features. Whitney (2004) and Adams (1999) outlined and implemented algorithms necessary for the motion and constraint analysis of assemblies constructed by combining parts and using assembly features. (For details, see Appendix A.2; "Feature Toolkit" for 17 joint types is also given in Appendix A.2.5.) The general form of a twist is;

$$TR = \begin{bmatrix} \omega_{1x} & \omega_{1y} & \omega_{1z} & \upsilon_{1x} & \upsilon_{1y} & \upsilon_{1z} \\ \omega_{2x} & \omega_{2y} & \omega_{2z} & \upsilon_{2x} & \upsilon_{2y} & \upsilon_{2z} \\ \omega_{3x} & \cdots & & & & \end{bmatrix} \tag{4.1}$$

(← each row for an independent d-o-f)

where ω is angular velocity, *v* is linear velocity, sub-index 1 is for the first d-o-f, and *x*, *y*,and *z* are primary axes.

Another matrix used for defining associations in the assembly is the wrench matrix, which is used for constraint analysis with flow of force and moment. A wrench is defined in terms of forces and moments in the following way:

$$WR = \begin{bmatrix} f_{1x} & f_{1y} & f_{1z} & M_{1x} & M_{1y} & M_{1z} \\ . & . & . & . & . & . \end{bmatrix} \qquad (4.2)$$

where "*f*" is force, "*M*" is moment, and *x*, *y*, and *z* stand for the directions. The wrench matrix fits the twist matrix, and it is composed of all the forces and moments that the joint is able to resist.

The relative positions and orientations of the features and artifacts (using the local coordinate system [LCS] for the artifacts and the feature coordinate system [FCS] for the features) are defined in chapter 3.

The next step is to define the connection (joint) properties of associated artifacts and assembly features in terms of twist matrices and frames of the assembly features, based on the degrees of freedom of the connections, and stored in AssemblyFeature, AssemblyFeatureAssociation, Connection, KinematicPair classes in the modified OAM model objects. When many parts are associated, one can know how by referring to the degree of freedom information, and one can know how any change in constraints will affect the functionality of mathematically related entities. Even though, estimating/assessing the effect of changes in behavioral model requires complete mathematical characterization of function and behavior. In this study, only the effect of degrees of freedom on behavior is discussed.

4.5.2 Motion and Constraint Analysis of Assembly Associations

A detailed motion and constraint analysis of assembly associations is given in Appendix 2.3.4. Screw theory-based representation makes possible motion limit analysis for behavior. Since assembly associations are connected mathematically, it is easier to define connections in the behavior model. For example, for a pin-and-hole connection, parametric assembly constraints like "coaxial axes" can be applied, and the d-o-f becomes 2 (1 rotational about z and 1 translational on z), and this connection would have 4 wrench matrices (force and moments). In the behavioral model, it is modeled that way. However, if any motion is constrained because of any other connection in the assembly, the reaction force and moments would, in actuality, differ from those defined in the behavioral model. So motion and constraint analysis is very important in the behavioral model and for tracing of problems in the system/assembly.

4.5.3 Calculating Combinations of Twists and Wrenches

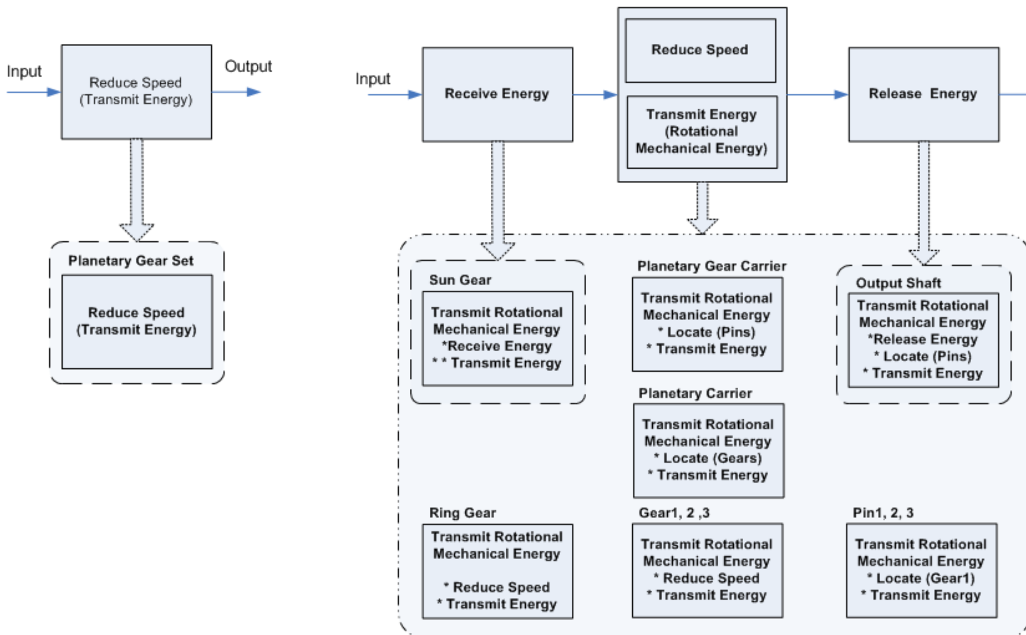
Between artifacts, there might be more than one assembly association that defines the degree of freedom and the twist and wrench matrices. In section 4.5.1, an assembly feature is described and its twist matrix identified. So now it is necessary to find a method to team up two or more assembly features and obtain the resulting twist matrix which describes the motions allowed by the grouping of the assembly features. To compute the intersection of the twists for each assembly feature association, an application of screw theory has been developed by Whitney (2004) and his students, available in Section 4.E.2.d.2 (Whitney, 2004). This method is applied for the calculation of combined assembly feature associations in an artifact association between two artifacts. As in the motion and constraint analysis, the mathematical combinations of twist matrices for every assembly feature association show the associations between the two artifacts. Combining all of the assembly feature associations between two artifacts will result in the artifact association between any two given artifacts. In other words, an assembly feature association has a degree of freedom (i.e. motion capability), and the combination of assembly feature associations between two artifacts gives the motion capability of the artifacts association between the same artifacts.

In the next section, interactions among associations of assembly, function and behavior in the product information representation and exchange model are discussed, and are illustrated using the planetary gear box example

4.6 Case Study

In this section, a functional and behavioral product information representation model is applied to the case of a planetary gear set.

First, the main function ("*Reduce* Speed") is decomposed into sub-functions ("*Receive* Energy", "*Transfer* Motion", "*Support* Load" and "*Release* Energy"), and then appropriate artifacts (e.g., a planetary gear set for the "*Reduce* Speed", and other sub-assemblies/parts for the sub-functions) are chosen for these functions (Figure 4.21). The functions of the artifacts, with input and output flow (energy, material, and signal), the power variables (of the energy flow), and the associations among artifact functions are shown in Figure 4.7. The entire function structure, including the lowest-level artifact functions and their sequence in the structure, is depicted in Figure 4.22, which also shows the flow components of functions, like Rotational Mechanical Energy (RME), which have angular velocity (ω) and torque (T); Translational Mechanical Energy (TME), which have force (f) and linear velocity (v) for the "*Transfer* Motion" function.



(a) (b)

Figure 4.21: Functional Decomposition and Assigning Artifacts

Second, the relative positions and orientations (P/O) of the parts and the frames of the features in the assembly (in terms of the local and feature coordinate systems) and the P/O of power couples' entry/exit (i.e., P/O of input force) are defined with transformation matrices. The connection information between two artifacts is transformed into twist matrices, based on the degree of freedom of the artifact (part or assembly), to enable tracking of the energy parameters (force, torque, and angular and linear velocity).

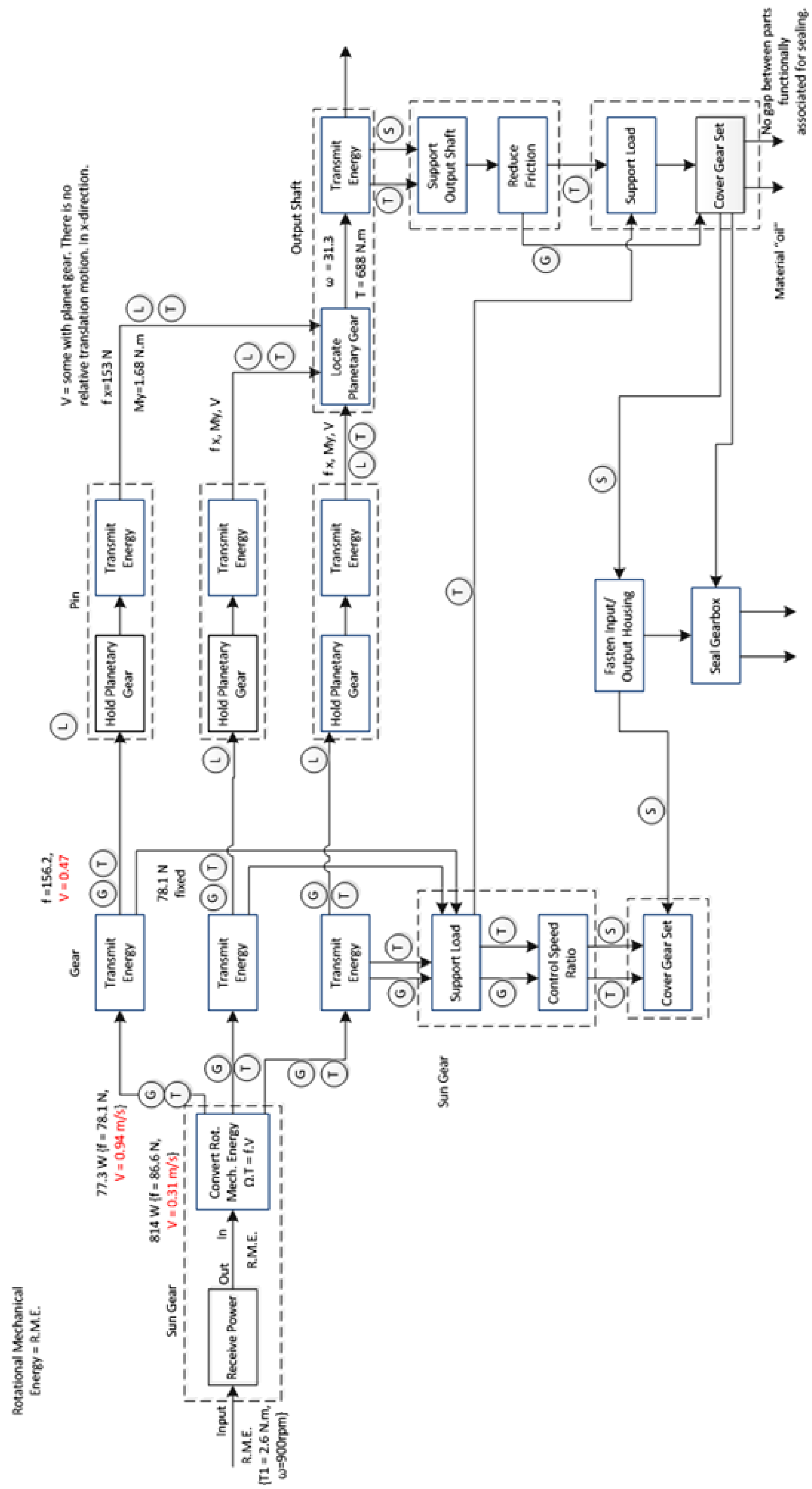


Figure 4.22: Interrelationships among Function, Behavior and Artifact

The combined transformation matrix between Pin1 and Gear1 (see Figure 4.23) is given as

$$T_{Pin1-Gear1} = T_{Pin1-Pin1AF} \cdot T_{Pin1AF-Gear1AF} \cdot T_{Gear1AF-Gear1} \quad (4.3)$$

Since the positions and orientations of the parts in the local coordinate systems and feature coordinate systems are the same, the transformation matrix becomes as follows:

$$T_{Gear1-Gear1AF} = T_{Pin1-Gear1AF} = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} . \quad (4.4)$$

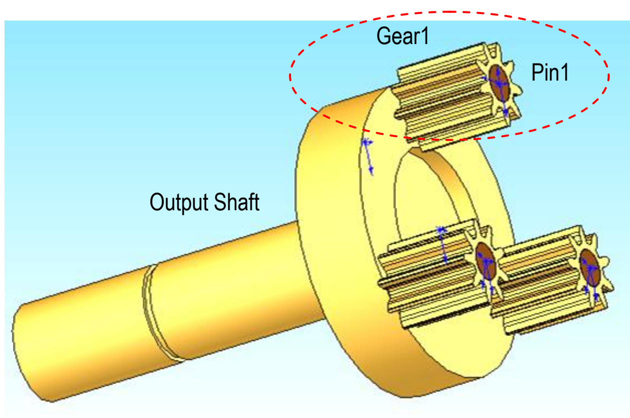


Figure 4.23: Planetary Gear Carrier Subassembly

There is also a different type of functional association, other than those based on spatial relationships, which is Maintaining Clearance between the Output Shaft and Gear1, under the loading shown in Figure 4.24, based on design requirements which are related to Gear1, Pin1 and the Output Shaft. The required information relating to the geometry, material properties and positions comes from the Form and Association classes in the OAM model (for example the Kinematic Pair Information of the Gear1-Pin1 Assembly in Figure 3.13).

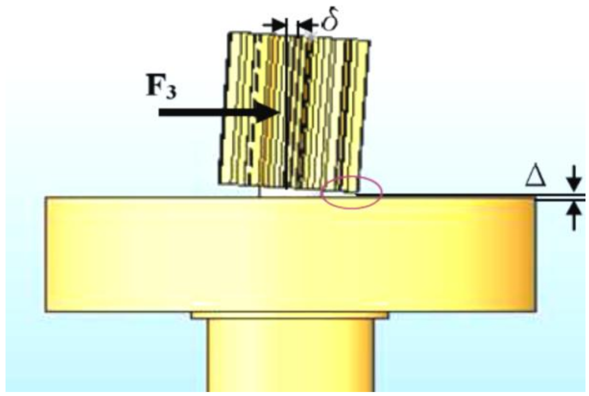


Figure 4.24: The Gap between the Output Shaft and the Planetary Gear after Loading

As mentioned above, there might be other types of functional associations between artifacts, based on design requirements. Here, the design requirement is the clearance Δ between Gear1 and the Output Shaft (Figure 4.24). So it is necessary to add a functional association for the artifacts Output Shaft Gear1 and Pin1, which will directly affect the clearance between Gear1 and the Output Shaft. The deflection (δ) on Pin1 will directly affect the position of Gear1.

Third is defining twist and wrench matrices, based on the associations with other artifacts. The behavioral model is developed based on functional associations (Transferring Force/Moment). The connection between Gear1 and Pin1 is a moveable connection; it has 2 degrees of freedom, which consist of rotation about the *x* axis.

TR = [0 0 1 0 0 0] *allows rotation about z-axis*

TR = [0 0 0 0 0 1] *allows translation on z-axis*

For constrained movements (translational and rotational) the wrench matrices are

w_1 = [0 0 0 0 r_x 0]	*support for the moment about y*
w_2 = [0 0 0 0 0 r_x]	*support for the moment about z*
w_3 = [0 1 0 0 0 0]	*support for the force along y*
w_4 = [0 0 1 0 0 0]	*support for the force along z*

For all constrained movements (translational and rotational), there will be support for, or reactions to, all of the forces coming from Gear1 and affecting Pin1. As a result of functional and artifact associations, which define constraints, the free body diagram (Figure 4.25) is built and it shows the forces and moments of the artifact Pin1.

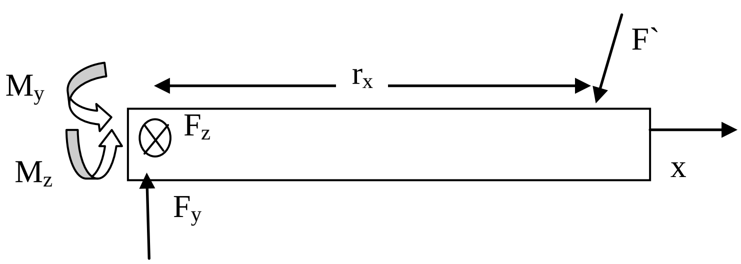


Figure 4.25: Free body diagram of Pin1

The behavior model for this functional association is developed based on the deformation of Pin1 under loading conditions. Since Pin1 is subject to a bending moment and should support this loading, the behavioral model is built with engineering formulas applicable to relevant functional associations; engineering formula for the deformations;

$$\delta = \frac{F\,L^3}{3\,E\,I} \quad where\ I = \frac{\pi\,d^4}{64} \tag{4.5}$$

for the bending stress;

$$\sigma = \frac{M\,c}{I}\ , M = F\ L \quad where\ I = \frac{\pi\, d^4}{64} \tag{4.6}$$

The other functional and behavioral information and the associations among function, behavior, and form are given in Figure 4.26. As can be seen from the figure, the artifact Gear1 has functions (Support Load, Transfer Motion) and different functional associations (Transferring Force/Moment, Maintaining Clearance, and Transferring Motion). For Transfer Motion, the input is rotational mechanical energy, and the parameters are angular velocity and torque. Since the functional association is Transferring Motion, based on the type of loading, the behavioral model is built with appropriate engineering formulas and physical laws. For the Supporting Force and Moment functional association, we have another behavioral model that has appropriate formulas. Behavior is then evaluated using all these behavioral models according to known inputs/effects. However, there might also be unknown effects from the environment. In that case, behavior other than the estimated behavior can be observed, and changed. There is also a different type of functional association, other than that based on spatial relationship: Maintaining Clearance between the Output Shaft and Gear1, under the loading shown in Figure 4.26, based on design requirements which are related to Gear1, Pin1 and the Output Shaft.

As can be seen from the Figure 4.26, in real life, behavior is not only about intended or foreseeable environmental conditions. It is observed that temperature is increased, which means, there is heat generation because of friction among associations. Heat flow is then added to the system where heat generation occurs, and, as a result, a new function, "Remove Heat," has to be added to the system to overcome the unintended behavior. This new function requires a new artifact or modification of an artifact to allow heat dissipation/removal (see Figure 4.27). Section 5.6 explains how to add or modify an object in the product information model.

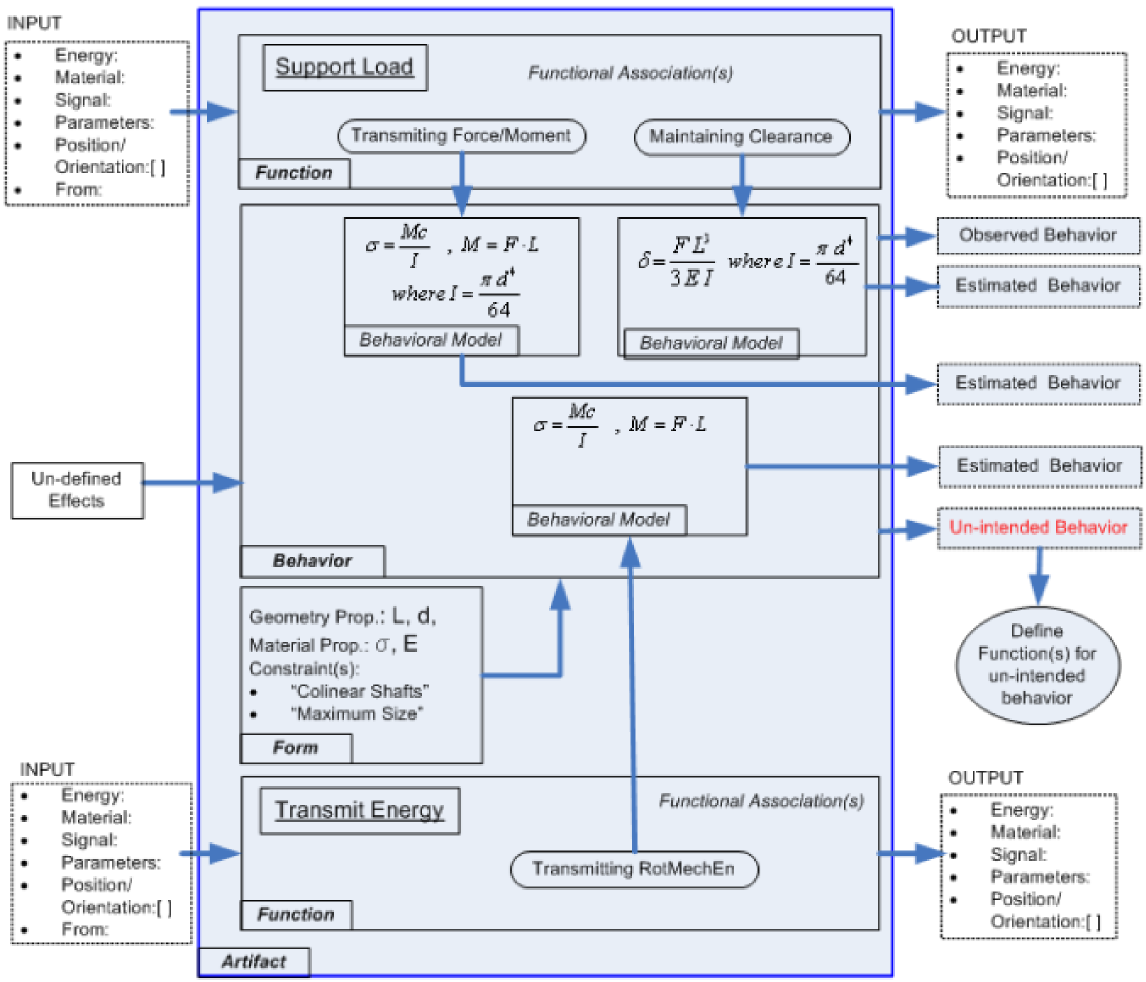


Figure 4.26: Interrelationships among the Function, Behavior and Form of the Artifact

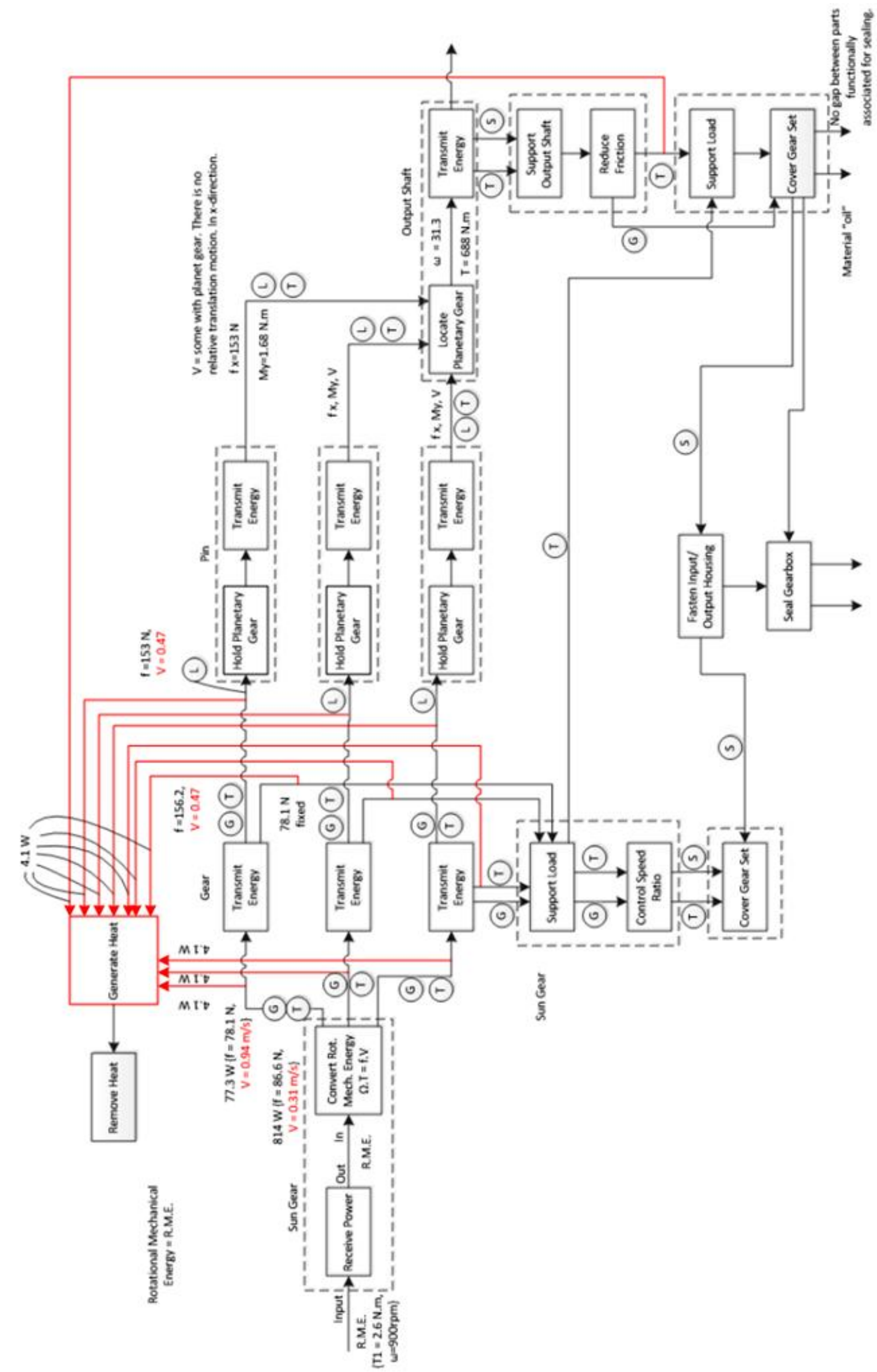


Figure 4.27: Adding a New Function into Diagram to overcome Un-intended Behavior

Chapter 5

Consistency Validation in the Product Information Model

5.1 Introduction

The development of the assembly-related product information representation will be foundation for exchange model, which will provide mechanisms to capture product information, store it in a database, and allow access to it. First phase is representation of functional and behavioral product information and definition and characterization of consistency validation rules, which ensure that product information will not contradict itself or be traceable through the associations without disconnection. The traceability feature also assists to find causes of failures by tracking it through relevant associations. The requirements for replacing a part or modifying a part are discussed. When there is a need for replacing the artifact with another one, one must consider all of the associations of existing artifact with other artifacts and environment, not just functional and space requirements, and the relevant modification(s) of the associated objects has to verified.

In this study, the second phase is to develop a framework and basis for an information exchange model with a tracing mechanism and a modification verification tool, based on the product information representation defined in chapter 3 and 4.

For the consistency of product information, repetitions of the same information in different places (classes and attributes) need to be minimized. If it is not possible to situate it in one place (e.g., the same axis information in feature geometry, assembly constraint, geometric tolerance, kinematic pair, etc.), the repeated information in different places in the information model has to be linked via rule-based constraints. The information model has to be complete in terms of the traceability of function, behavior, spatial relationships, etc., in order to support all information exchange activities. In this chapter, a brief introduction is given, followed by a discussion of the issues mentioned in chapter 3 and 4 about maintaining consistency in the product information model, and appropriate consistency validation rules are defined.

The original and modified versions of the CPM and OAM model were developed in UML, but for consistency and the completeness of the product information in terms of traceability, Ontological Web Language (OWL) has more capability. Mapping from the UML model to OWL (Fiorentini, 2007) is necessary; issues involving mapping are given in Appendix 4. Also, in the literature, some efforts have been reported on direct OWL-based product information representation, especially the work done by Kim et.al. (2006). The consistency validation rules need to be transformed into OWL constraints and Semantic Web Rule Language (SWRL) rules to be able use in OWL based tools. It is not only about translating validation rules into OWL constraints and SWRL rules; also it has to comply with the logic of OWL.

In the last section, the requirements for replacing or modifying a part are discussed. When there is a need for replacing the artifact with another one, one must consider all of the associations of the existing artifact with other artifacts and the environment, not just it's functional and space requirements.

5.2 Consistency and Validation Rules for the Modified CPM and OAM

Every artifact and feature has its own coordinate system, the relative positions and orientations of the artifacts, the features and ports of these artifacts are first defined with transformation matrices. A transformation matrix enables the calculation of the relative position and orientation of an entity with respect to others. Several transformations can easily be chained by multiplying the corresponding matrices. Second, the connections/assembly constraints/joints between parts and between features are defined, based on degrees of freedom, by applying screw theory (chapter 4). Third, the relations among these association classes are defined for different levels and perspectives (i.e., assembly constraints, kinematic pairs, geometric tolerances, etc.). All of these matrix-based definitions for positions/orientations, connections, and assembly constraints, along with the functional inputs/outputs and the behavioral model, which are described in chapter 4, are then utilized to develop a mathematical model.

5.2.1 Consistency Rules for Assembly Associations

As a backbone of the CPM and the OAM, valid definitions of associations are very important for a consistent and complete product information representation and exchange model. Kim et al. (2006) represent the relational constraints for the joints and the mating process in the assembly as follows:

Rule: If {AF1.hasBasicGeometry = Cylindrical AND AF2. hasBasicGeometry = Cylindrical AND {{AF1.Cylindrical.Type is Male} AND {AF2.Cylindrical.Type = Female}} AND {AFA1.hasAF AF1 AND AF2} AND {AFA1.ParametricAssemblyConstraint = “Coaxial”} THEN {AFA1.KPair = “CylindricalP”}

Rule: If {AF1.hasBasicGeometry = Cylindrical AND AF2. hasBasicGeometry = Cylindrical AND {{AF1.Cylindrical.Type is Male} AND {AF2.Cylindrical.Type = Female}} AND {AFA1.hasAF AF1 AND AF2} AND {AF3.hasBasicGeometry = “Planar” AND AF4. hasBasicGeometry = “Planar” AND {AFA2.hasAF AF3 AND AF4}{AFA2.ParametricAssemblyConstraint = “ParalellWithDim”} THEN {AFA2.KPair = “RevoluteP”}

5.2.2 Parametric Assembly Constraints and Geometric Tolerance Relationships

In order to define relationships, whether it be between different artifacts in an assembly or between the geometry information (on three levels) of the same artifact, the establishment of “rules/constraints” becomes necessary. For example, it is not possible to assign "Cylindricity" or "Circularity" geometric tolerances on a planar surface or "Flatness" tolerance to a cylindrical surface. This section describes what tolerance can be assigned to what type of geometry. Some validation rules for the tolerance class, which are adopted from Hu and Peng (2011), have been applied to the modified OAM. The following are the validation rules for the features listed:

IF { Feature.BasicShape = Spherical}, THEN {SizeTolerance AND/OR Circularity}.

IF {Feature.BasicShape = Cylindrical}, THEN {SizeTolerance AND/OR (Circularity OR Cylindricity)}.

IF {Feature.BasicShape = Planar}, THEN {flatness OR Straightness}.

IF {Feature.BasicShape = Cylindrical, PAC = Perpendicular, Datum = Plane}, THEN {Perpendicularity}.

IF {Datum= Straight line, Feature.BasicShape = Cylindrical, PAC= Parallel}, THEN {Parallelism and Distance-tolerance}.

IF { Feature.BasicShape = Planar, PAC= Parallel, Datum = Straight_line or Plane}, THEN {Position and Parallelism}.

5.2.3 Parametric Assembly Constraint Realizer and DOF Relationship

Each assembly constraint implies reduction of d-o-f. If two artifacts' assembly features are associated by FixedConnectors (e.g., pressfit, welding or bolt-nut) the degrees of freedom become zero (for more types see Table 3.4). For example:

Rule: If {{AFA.hasAF(Art1.AF1 AND Art2.AF2) }AND {ParametricAssemblyConstraintRealizer = "FixedConnector"}}, THEN {AFA.AFAhasDOF is "0"} AND {ArtA.ArtA.ArtAhasDOF is "0"} AND {Connection.Type = Fixed}.

5.2.4 Relationships among Connection, Assembly Constraints and Kinematic Pairs

As we mentioned in Section 3.2.2, the Connection type of artifact association depends on the combined degrees of freedom, which is calculated from all of the assembly constraints between assembly features of the same artifacts. At the same time, the combination of assembly constraints between assembly features of theses artifacts may define a type of kinematic pair. For example, the coaxiality assembly constraint between two cylindrical surfaces (one a hole and the other a cylinder) gives us a cylindrical pair. Both have the same degree of freedom in translational and rotational movability along the axis. The requirements are as follows:

- The basic shape of both assembly features has to be a cylindrical surface
- One must be a hole, the other must be a cylinder
- The fit type must be clearance
- The axis of cylindrical surfaces must be aligned (coaxial).

Rule: If {AF1.hasBasicGeometry = Cylindrical AND AF2. hasBasicGeometry = Cylindrical AND {{AF1.Cylindrical.Type is Male} AND {AF2.Cylindrical.Type = Female}} AND {AFA1.hasAF AF1 AND AF2} AND

{AFA.ParametricAssemblyConstraint = “Insert”} THEN {AFA1.AFAhasDOF = [0 0 1 0 0 1]}

SWRL: AFhasBasicGeometry (?x, Cylindrical) Λ AFhasBasicGeometry (?y, Cylindrical) AFhasBasicGeometryType (?x, male) Λ AFhasBasicGeometryType (?y, female) Λ AFAhasAF(?z,?x) Λ AFAhasAF(?z,?y) Λ AFAhasParametricAssemblyConstraint (?z,Insert) → AFAhasDOF([0 0 1 0 0 1])

5.2.5 Effect of Tolerances and Fit Types on Degree of Freedom

The tolerance fit type will affect the degrees of freedom by physically preventing motion. For example, when the tolerance value of the diameter of the hole or the cylinder is modified, and the fit type becomes “interference,” then this connection turns into a fixed connection. Therefore, relevant consistency rules have to be defined in a mathematical model. The ParametricAssemblyConstraint is related to geometric tolerances by the TolerancedAssemblyConstraint subtype of the ParametricAssemblyConstraint. For example, the following is a pin-hole assembly association between two assembly features:

Rule: If {AF1.hasBasicGeometry = Cylindrical AND AF2. hasBasicGeometry = Cylindrical AND {{AF1.Cylindrical.Type is Male} AND {AF2.Cylindrical.Type = Female}} AND {AFA1.hasAF AF1 AND AF2} AND {FitType = “Clearence”} AND {AFA.ParametricAssemblyConstraint = “Insert”} THEN {AFA1.AFAhasDOF = [0 0 1 0 0 1]}

SWRL: AFhasBasicGeometry (?x, Cylindrical) Λ AFhasBasicGeometry (?y, Cylindrical) AFhasBasicGeometryType (?x, male) Λ AFhasBasicGeometryType (?y, female) Λ AFAhasAF(?z,?x) Λ AFAhasAF(?z,?y) Λ AFAhasParametricAssemblyConstraint (?z,Insert) →AFAhasDOF([0 0 1 0 0 1])

On the other hand, if the tolerance between the pin and the hole for the insert constraint is an interference fit, which constrains all of the degrees of freedom, then this association becomes a fixed connection with d-o-f [0 0 0 0 0 0].

Rule: If {AF1.hasBasicGeometry = Cylindrical AND AF2. hasBasicGeometry = Cylindrical AND {{AF1.Cylindrical.Type is Male} AND {AF2.Cylindrical.Type = Female}} AND {AFA1.hasAF AF1 AND AF2} AND {FitType = "Interference"} AND {AFA.ParametricAssemblyConstraint = "Insert"} THEN {AFA1.AFAhasDOF = [0 0 0 0 0 0]}

5.2.6 Validation for Function and Flow Properties

Function types prescribe many restrictions to flows such as validation for function classification and input-output flow existence. For instance, function "*Transmit*", which is a "Tranfer Function", must include input and output flows without any differentiation about the flow type, whereas the function "*Contain*", which is a "Store Function" needs to have input flow but not output flow. Similarly, "Supply Function" needs to have output flow but not input flow.

Validation for Function Classification and Input-Output Flow Existence

TransferFunction:

Rule: If {Function.Type = TransferFunction} THEN {Function hasInputFlow and hasOutputFlow}

SWRL: hasInputFlow (?x,?y) Λ hasOutputFlow (?x,?z) → TransferFunction (?x)

StoreFunction:

StoreFunction, by definition, has no output flow. However, SWRL does not support negated atoms. The concept will be defined in OWL by constraints. Two constraints need to be defined. The first consists of the necessary conditions; the second, of the necessary and sufficient conditions.

Rule: If {Function.Type = StoreFunction} THEN {Function hasInputFlow } (not hasOutputFlow)

Necessary Conditions for StoreFunction:

OWL Restriction: hasInputFlow someValuesFrom Flow

This means that there should be at least one hasOutputFlow object property connected to a Flow.

Necessary and Sufficient Conditions for StoreFunction:

OWL Restriction: hasOutputFlow max 0

This means the maximum number of hasInputFlow object properties can be 0.

SupplyFunction

SupplyFunction, by definition, has no input flow. However, SWRL does not support negated atoms. The concept will be defined in OWL by constraints. Two constraints need to be defined. The first consists of the necessary conditions; the second, of the necessary and sufficient conditions.

Rule: If {Function.Type = SupplyFunction} THEN {Function hasOutputFlow} (not hasInputFlow)

Necessary Conditions for SupplyFunction : This means that there should be at least one hasInputFlow object property connected to a Flow.

OWL Restriction: hasInputFlow someValuesFrom Flow

Necessary and Sufficient Conditions for StoreFunction :

OWL Restriction: hasOutputFlow max 0

This means the maximum number of hasOutputFlow object properties can be 0.

Validation for Function Type and Input/Output Flows

Validation for function type and input/output flows can be developed as following;

Rule: If {Function = "Change"} THEN {InputFlow.TypeOfFlow = OutputFlow.TypeOfFlow}

SWRL : hasInputFlow (?x,?y) Λ hasOutputFlow (?x,?z) Λ flowHasType (?y,?y1) Λ flowHasType (?z,?z1) Λ differentFrom (?y1,?z1)→functionHasType(?x,Change)

Rule: If Function is “Convert” → InputFlow.TypeOfFlow ≠ OutputFlow.TypeOfFlow

SWRL : hasInputFlow (?x,?y) Λ hasOutputFlow (?x,?z) Λ flowHasType (?y,?y1) Λ flowHasType (?z,?z1) Λ sameAs (?y1,?z1) → functionHasType(?x,Convert)

Other function types in the taxonomy in Table 4.1 can be defined in a similar way.

5.3 Verification of Replacing/Modifying an Artifact in the Assembly

When there is a need to replace an artifact, one must check the associations and specifications in the base artifact against the ones in the candidate artifact. We extended the CPM information model to incorporate more information about the associations and specifications of the “base” artifact, in terms of functional and assembly associations, through the “ports,” which are special features in the OAM. These specifications include: (i) assembly associations—the basic shape, the position and orientation of assembly features, and the connection type (including kinematic pair information, if it exists), and (ii) functional associations—the function, the required input/output energy/material/signal (including the positions and orientations of forces, moment, if it exists) and the design requirements (e.g., maintain clearance, etc.).

In this modified CPM, artifacts are defined from different perspectives (Function, Behavior and Assembly Structure) and at different levels of abstraction (e.g., Association, Assembly, Artifact and Feature). When there is a need to replace one artifact with another, one must compare all of the associations of the existing artifact with other artifacts and with the environment, not just address the functional and spatial requirements. For any end of lifecycle (EOL) operation (a replacement process, for example), if a search of existing (old model) artifacts for the required product is needed, higher-level requirements like functions and sub-functions are searched first. Functional inputs and outputs (energy, material signal) are then checked. Next, the structural specifications, like the overall size, the basic shape, the position and orientation of the features, the type of connection (including kinematic pair information, if there is any), and so forth, are checked. Lastly, the lowest level abstractions of structure (e.g., feature of size) and function (e.g., force, moment, velocity in transferring mechanical energy) are checked if the replacement part fits.

In this section, the usage an existing artifact from the company's model database is studied to replace the sun gear in the planetary gear box.

Figure 5.1 shows summary information about the *sun gear* artifact. It includes function(s), assembly features/ports, the basic shape, the assembly/artifact/assembly feature associations and functional associations. For the "Reduce Speed" function, the database should be searched for appropriate candidates. In this case, another sun gear (Sungear2) matches the requirements. Then, for the functional association "Reduce Speed", input/output torque and angular velocity (gear ratio) information is checked and found acceptable. The related assembly feature association representation "GearPair" information (including, the radius of the first link, the radius of the second link, the bevel angle, the helical angle, the gear ratio and the module) gives us a few more details to check. This detailed level information is given in Figure 5.2. The specifications of the replacement sun gear (Sungear2) match those specifications. On the other hand, the sub-function of the Transfer Energy function requires functional and assembly associations in port1. Here, even though the functional association is provided by Sungear2, one of the assembly feature associations is not satisfied. Sungear2 does not have the necessary assembly feature (an axial slot) to match the assembly feature association for the "Receive Energy" functional association. The diameter of the shaft (CylindricalSurface) assembly feature is also larger than what is required. Therefore, one needs to develop remanufacturing strategies to satisfy the remaining specifications, like reducing the shaft diameter, adding a slot feature, etc., as needed (Figure 5.3).

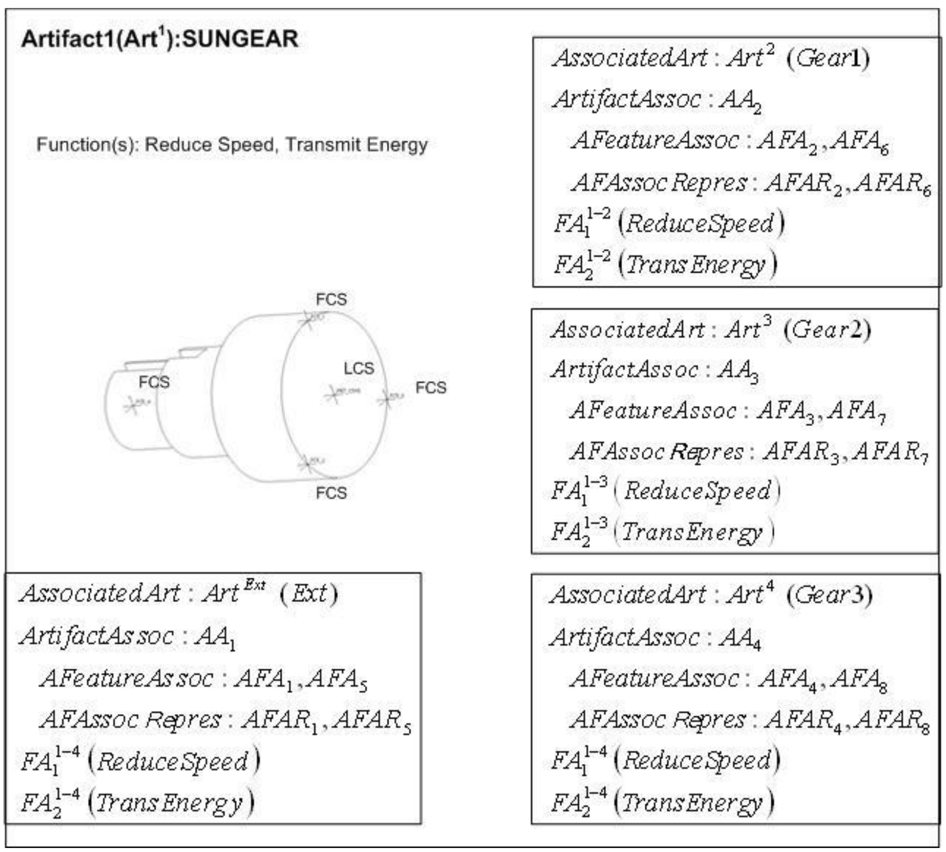


Figure 5.1: Artifact Information about the Sungear

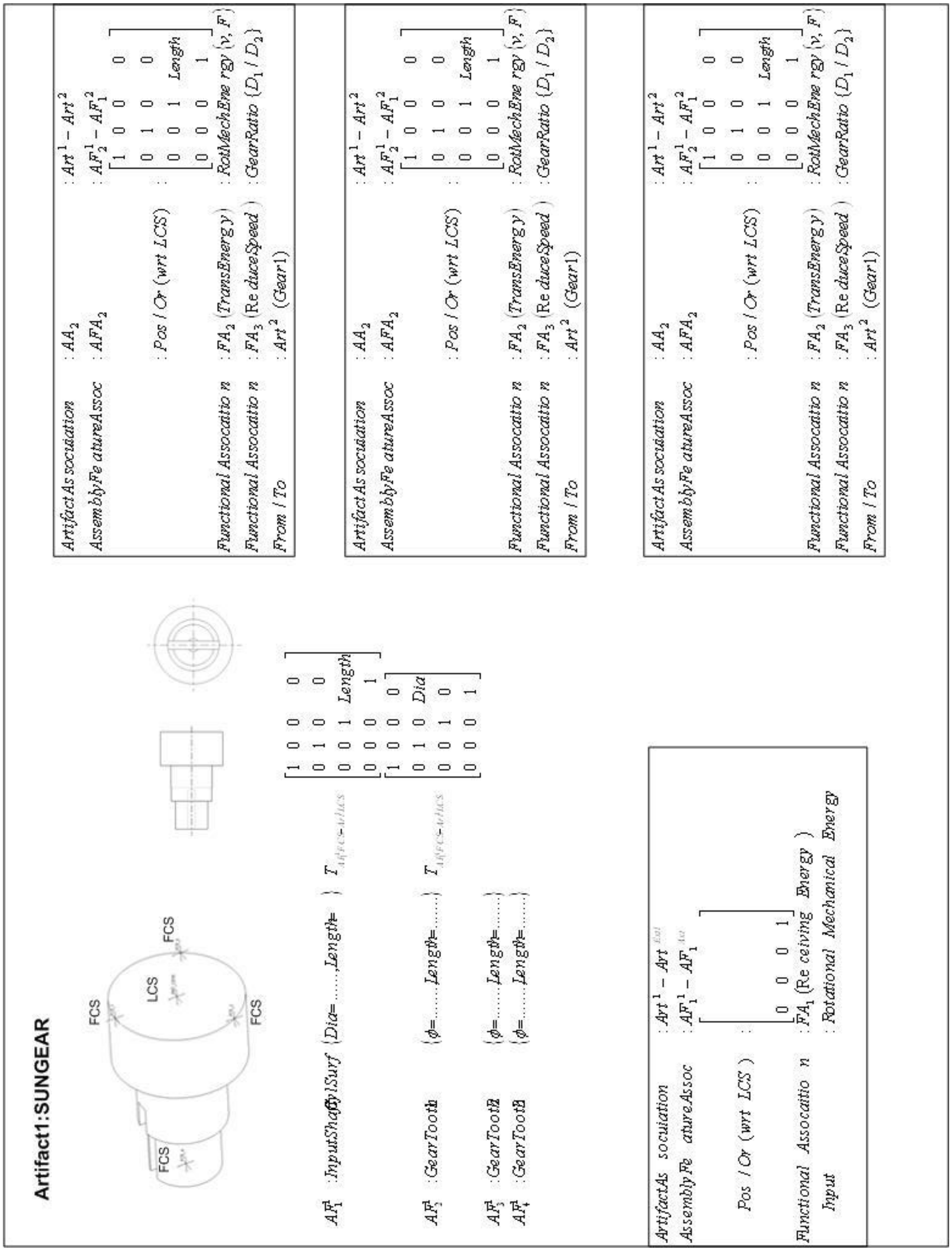


Figure 5.2: Assembly Feature- Level Information for the Sungear with Ports

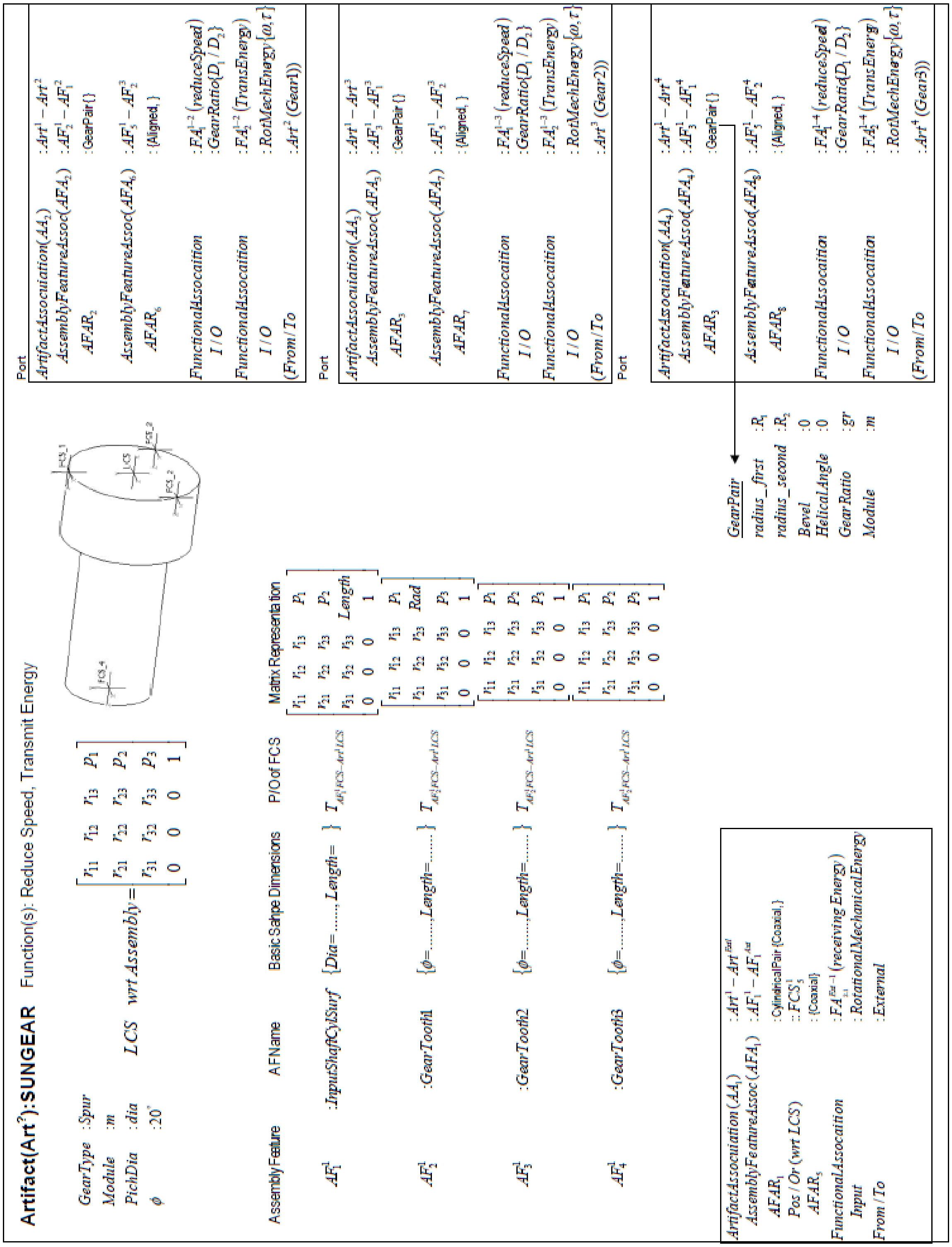


Figure 5.3: Feature- Level Information for the Replacement Artifact for the Sungear

Chapter 6

Conclusion and Future Studies

6.1 Conclusion

The main objective of this research is to develop an assembly-related product information representation that takes into account product function and behavior, in order to provide a mechanism for exchanging product information throughout the lifecycle of a product. This will enable efficient collaboration among different stakeholders, reduce interoperability costs, and reduce product development time. In chapter, we mentioned five objectives of this research study. Those objectives and related accomplishments will be discussed in details in the following paragraphs. The objectives are given in chapter 1, and contributions made in relation to those objectives are mentioned below.

Objective#1: To develop an assembly-related product information representation that includes assembly structure, spatial and design relationships, and the connection/joint properties of associated artifacts and features.

Issues, related to this objective, involved interconnections among classes in the OAM model, usage of the same product information in different classes and at different levels of abstraction, and the aggregation of lower-level information to higher levels. The main component in the OAM is the defining of “associations” among artifacts and artifacts’ features. The spatial relationships in the assembly and the other connection and joint properties (i.e., degrees of freedom) of associated artifacts and artifacts’ features were described in this work. Also, details about classes, like attributes and associations with other classes, were defined. On the other hand, the model needs to be extended to represent complicated geometries and detailed geometric information like in Boundary Representation (B-Rep).

The CPM/OAM product information model was extended to incorporate more information about the associations and specifications of a “base” artifact, in terms of functional and assembly associations through the “ports” that are special features in the OAM. These specifications include assembly associations—the basic shape, the position and orientation of assembly features, and the connection type (including kinematic pair information, if it exists).

Objective#2: To define the assembly structure and associations, with mathematical characterization, to make the assembly model consistent, correct, and complete in terms of traceability.

In this study, we developed the mathematical definition of interrelationships in the extended OAM model. These relationships are:

- assembly feature associations and parametric assembly constraints (screw theory for assembly constrains is applied).
- artifact associations, as aggregations of assembly feature associations (a combination of twist matrices of screw theory is applied)
- parametric assembly constraints and kinematic pairs
- degrees of freedom of AFA and ArtAs
- effects of tolerance (fit types) on degrees of freedom
- basic shape and tolerance
- parametric assembly constraints and tolerance
- connection and parametric assembly constraints.

Parametric assembly constraint realizers (as can be understood from the name) are introduced for physical realization of assembly constraints, either through special connectors or through processes like welding. That is the connection between assembly associations and behavior.

The mathematical definitions for Intermittent Connection, Kinematic Path, and Position Orientation as specializations of Artifact Associations are still need to be addressed because of the complexity of the problem and limited usage.

Objective#3: To develop functional and behavioral models to define the interrelations among the function, behavior and form of an artifact throughout the product development stages. These interrelations not only involve input/outputs (e.g., *output speed and input speed), but also relations (associations among an artifact's spatial and design relations).*

Issues related to Objective #3 involve:

- the use of different function and flow terminology in the literature
- the development of function structure (decomposition and sequence)
- the relationship between function type and input/output flow
- the connection of functions with artifacts
- the definition of interconnections among function, artifact and behavior.
- the definition of behavior and its specializations
- the definition and integration of unintended behaviors
- the introduction of new functions into the system
- the addition of FMEA to the behavioral model
- the definition of functional associations to trace what artifacts are related to what function or what types of associations exist among artifacts, etc.

In the functional model, the first function to be defined is the overall function. This is then decomposed into, and supported by, sub-functions. Each function has a certain priority. In this way, it is assured that a certain function has to wait until the prior one(s) is processed. Another property is functional associations, which define the relationships among the artifacts and the behavior of the artifacts based on these associations. The more important feature of the model developed in this work is that function, behavior and assembly information are interrelated. Function and behavior are related through functional associations, and behavior and artifact are related through the behavioral model. Therefore, if any change is required or a problem occurs, a designer will be able to (1) check the intention behind that feature, artifact or any artifact association in the assembly, (2) see how it will affect the other entities, and (3) trace any problem through the associations.

Objective#4: To develop an assembly-related product information representation as a foundation for mechanisms to capture product information. It should allow users to browse, edit, and transfer product information with a verification tool that will check the consistency of the modified information.

The mapping of data from the UML-based CPM and OAM into OWL requires special expertise. Direct modeling in OWL is easier, but because of its ontology it does not have *n*-ary associations, but rather binary associations, which creates problems when one is defining assembly and functional structures, since they might have subassemblies of subassemblies of subassemblies. There has been research efforts on mapping UML based model into OWL, but there is no solution yet.

Most rules for relationships between relevant classes (and attributes) are defined, except the ones peculiar to an individual product. SWRL rules have to be generic and correct for every case, but some of the consistency rules require individual definition, since they may involve a variety of entities.

Objective#5: To define a basis for a tracing mechanism that checks the consistency of the information and finds the causes of failures by tracking through associations.

For consistency, the use of the same product information in different classes and at different levels of abstraction, and the aggregation of lower-level information to higher levels, were the problems that have been solved by connecting the relevant information in different places through constraints and rules. In addition, while modifying any part of the product information associated with other objects, the relevant modification(s) of the associated objects has to be approved by a verification process defined by the SWRL rules and the OWL constraints defined in the OWL-based product information model. As mentioned in the discussions above, traceability is another consideration in an information model; it is provided for by interrelating objects (functions, behaviors, features, etc.) in a way that enables users to follow an object from the main functional requirements to the sub-functions and from the artifact to the design rationale arguments. Completeness, another important consideration, is defined in terms of traceability, so that there is no deficiency or disconnection among these entities in the system. If an information model is to support all information exchange activities, it has to be complete in terms of the traceability of function, behavior, spatial relationships, and so forth.

In the modified CPM, artifacts are defined from different perspectives (function, behavior and assembly structure) and at different levels of abstraction (e.g., association, assembly, artifact and feature). When there is a need to replace one artifact with another, one must compare all of the associations of the existing artifact with other artifacts and with the environment, not just address the functional and spatial requirements. For any end of lifecycle (EOL) operation (a replacement process, for example), if a search for existing (old model) artifacts for the required product is needed, higher-level requirements like functions and sub-functions are searched first. Functional inputs and outputs (energy, material signal) are then checked. Next, the structural specifications, like the overall size, the basic shape, the position and orientation of the features, the type of connection (including kinematic pair information, if there is any), and so forth, are checked. Lastly, if the replacement part fits, the association specifications for the geometry-level assembly features (which are the lowest-level abstraction of functional associations like force, moment, and velocity in transferring mechanical energy) are checked.

In summary, to foster an effective collaboration during product lifecycle activities, product information must include data on geometry and topology, assembly constraints and associations, design and product processes, the functions and behaviors of the product, and the design intent. There have been many efforts to connect function and behavior to structure, but there is no complete, consistent method yet.

This work should help people to make intelligent decisions by allowing them to manage product lifecycle activities from different perspectives (i.e., function, structure, etc.) using the knowledge of how the product information is interconnected, and how artifacts affect each other.

6.2 Future Studies

The long-range goal is to develop a representation and exchange model for general product information, encompassing all product lifecycle activities, which can be applied to most electro-mechanical products. Based on the proposed extended CPM and the

OAM, further efforts should be made to extend the research in these areas of (i) integration of the Failure Mode and Effect Analysis (FMEA) with the function – behavior model of the artifact, (ii) mathematical characterization of the function structure and functional associations, and (iii) the mapping of Unified-Modeling Language (UML) based CPM/OAM models into Ontology Web Language (OWL).

The FMEA technique can be integrated to define the behavior of an artifact based on predictions and possible failure modes and their effects. Just as this model will help with tracing problems in the product, it will also help the designer to define evaluation criteria for artifact selection. In this way, the designer will have parameters for checking how the environment or any other changes in conditions can affect the behavior of an artifact.

The mathematical characterization of associations can be extended to the function structure and functional associations, like defining the relationships between input and output flows by using the Law of Conservation of Energy.

The mapping of UML to OWL can be extended and completed to provide consistent and complete product information in terms of traceability, by using the consistency validation rules defined in chapter 5.

Appendices

Appendix – 1. The Original CPM and OAM Models

A-1.1. Overview of the Original CORE Product Model (Fenves, 2001)

The NIST Core Product Model (CPM) is a Unified Modeling Language (UML) based model intended to capture the full range of engineering information commonly shared in product development (Therani and Tanniru, 2005; Fenves, 2001). It consists of a set of classes, associations and class associations. In order to make the representation as robust as possible, the CPM is limited to a canonical set of attributes required to capture generic product information and to create relationships among them. The representation intentionally excludes attributes that are domain-specific (e.g., attributes of mechanical or electronic devices) or object-specific (e.g., attributes specific to function, form or behavior). A UML class diagram of the core product model is shown in Figure 1. In the text that follows, names of classes are capitalized (e.g., Information) and names of attributes are not (e.g., information). The classes comprising the CPM are grouped below into four categories: abstract classes, object classes, relationship classes and utility classes. Five abstract classes are used as base classes for other CPM classes: *CoreProductModel* represents the highest level of generalisation; all CPM classes are specialised from it according to the class hierarchy presented in Figure A1. The common attributes *type*, *name* and *information* for all CPM classes are defined in this class. *CommonCoreObject* is the base class for all the object classes. *CommonCoreRelationship* and its specialisations, the *EntityAssociation, Constraint, Usage* and *Trace* relationships, may be applied to instances of classes derived from this class. *CommonCoreRelationship* is the base class from which all association classes are specialized. It also serves as an association to the *CommonCoreObject* class. *CoreEntity* is an abstract class from which the classes *Artifact* and *Feature* are specialised. *EntityAssociation* relationships may be applied to entities in this class. *CoreProperty* is an abstract class from which the classes *Function, Flow, Form, Geometry* and *Material* are specialized. *Constraint* relationships may be applied to instances of this class.

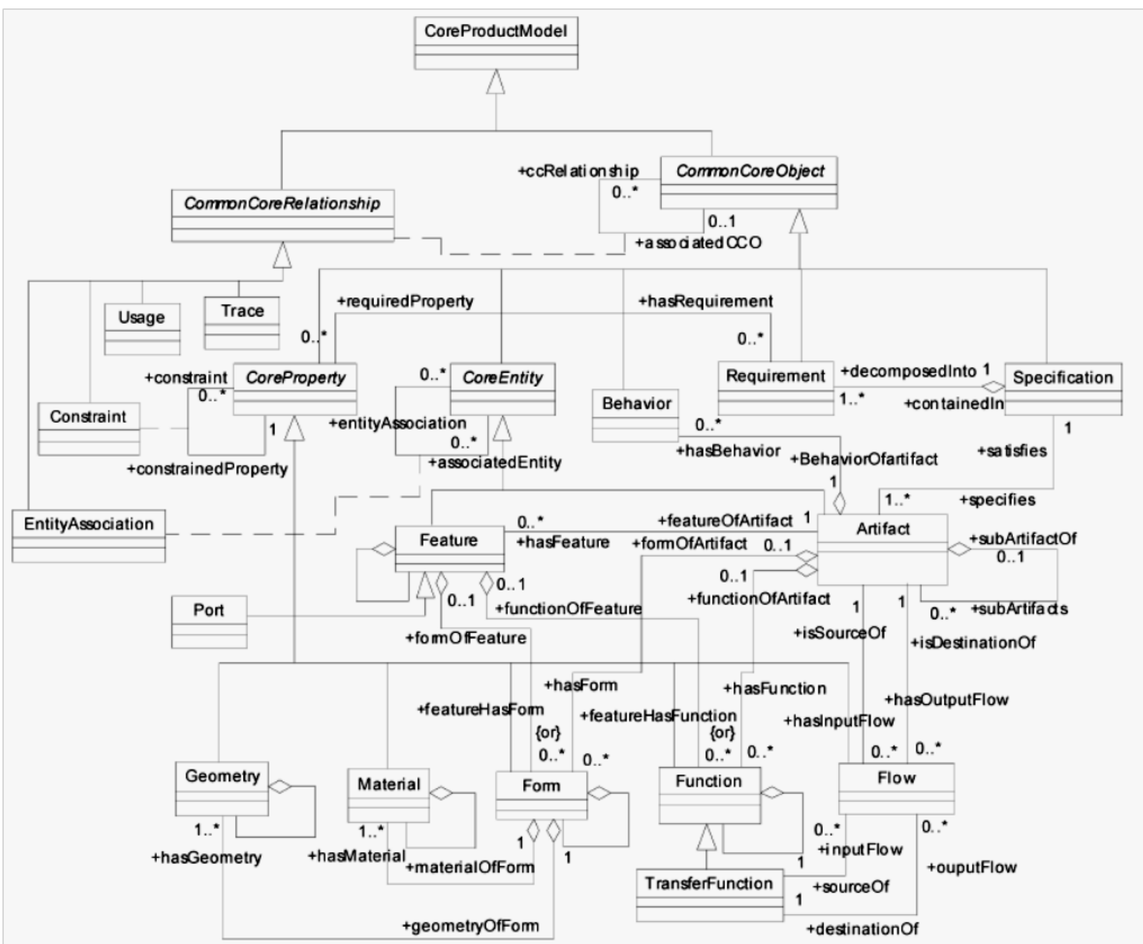


Figure A.4 Class diagram of the core product model (CPM)

The key object class in the CPM is the *Artifact*. *Artifact* represents a distinct entity in a product, whether that entity is a component, part, subassembly or assembly. All the latter entities can be represented and interrelated through the *subArtifacts/subArtifactOf* containment hierarchy. The *Artifact*'s attributes, refer to the *Specification* responsible for the *Artifact*, the *Form, Function* and *Behavior* objects comprising the *Artifact*, i.e., in UML terminology, forming an aggregation with the *Artifact*, and the *Features* comprising the *Artifact*. A *feature* is a portion of the artifact's form that has some specific function assigned to it. Thus, an artifact may have design features, analysis features, manufacturing features, etc., as determined by their respective functions. *Feature* has its own containment hierarchy, so that compound features can be created out of other features (but not artifacts).

A *port*, a specialization of *Feature,* is a specific kind of feature (sometimes referred to as an interface feature) through which the artifact is connected to (or interfaces with) other artifacts. The semantics of the term 'port' are deliberately left vague; in some contexts, ports only denote signal, control or display connection points, while in other contexts, ports are equivalents of assembly features through which components mate. A *specification* represents the collection of information relevant to an *Artifact* deriving from customer needs and/or engineering requirements. The *Specification* is a container for the specific requirements that the function, form, geometry and material of the artifact must satisfy. A *requirement* is a specific element of the specification of an artifact that governs some aspect of its function, form, geometry or material. Conceptually, requirements should only affect the function, i.e., the intended behavior; in practice, some requirements tend to affect the design solution directly, i.e., the form, geometry or material of the artifact. Requirements cannot apply to behavior, which is strictly determined by the behavioral model.

A.1.2. The Original Open Assembly Model (Sudarsan et. al., 2003)

Most electromechanical products are assemblies of components. The aim of the Open Assembly Model (OAM) is to provide a standard representation and exchange protocol for assembly and system-level tolerance information. OAM is extensible; it currently provides for tolerance representation and propagation, representation of kinematics, and engineering analysis at the system level. The assembly information model emphasizes the nature and information requirements for part features and assembly relationships. The model includes both assembly as a concept and assembly as a data structure. For the latter it uses the model data structures of ISO 10303, informally known as the STandard for the Exchange of Product model data (STEP).

Figure A.5 shows the main schema of the Open Assembly Model. The schema incorporates information about assembly relationships and component composition; the former is represented by the class **AssemblyAssociation** and the latter is modeled using part-of relationships. The class **AssemblyAssociation** represents the component assembly relationship of an assembly. It is the aggregation of one or more **Artifact Association**s.

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.

An **Assembly** is decomposed into subassemblies and parts. A **Part** is the lowest level component. Each assembly component (whether a sub-assembly or part) is made up of one or more features, represented in the model by **OAMFeature**. The **Assembly** and **Part** classes are subclasses of the CPM **Artifact** class and **OAMFeature** is a subclass of the CPM **Feature** class.

Artifact Association is specialized into the following classes: PositionOrientation, RelativeMotion and Connection. PositionOrientation represents the relative position and orientation between two or more artifacts that are not physically connected and describes the constraints on the relative position and orientation between them. RelativeMotion represents the relative motions between two or more artifacts that are not physically connected and describes the constraints on the relative motions between them. Connection represents the connection between artifacts that are physically connected.

Connection is further specialized as **FixedConnection, MovableConnection,** or **IntermittentConnection**. **FixedConnection** represents a connection in which the participating artifacts are physically connected and describes the type and/or properties of the fixed joints. **MovableConnection** represents the connection in which the participating artifacts are physically connected and movable with respect to one another and describes the type and/or properties of kinematic joints. **IntermittentConnection** represents the connection in which the participating artifacts are physically connected only intermittently.

OAMFeature has tolerance information, represented by the class **Tolerance**, and subclasses **AssemblyFeature** and **CompositeFeature**. **CompositeFeature** represents a composite feature that can be decomposed into multiple simple features. **AssemblyFeature,** a sub-class of **OAMFeature,** is defined to represent assembly features. Assembly features are a collection of geometry 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.

The class **AssemblyFeatureAssociation** represents the association between mating assembly features through which relevant artifacts are associated. The class **ArtifactAssociation** is the aggregation of **AssemblyFeatureAssociation**. Since associated artifacts can have multiple feature-level associations when assembled, one artifact association may have several assembly features associations at the same time. That is, an artifact association is the aggregation of assembly feature associations. Any assembly feature association relates in general to two or more assembly features. However, as in the special case where an artifact association involves only one artifact, it may involve only one assembly feature when the relevant artifact association has only one artifact.

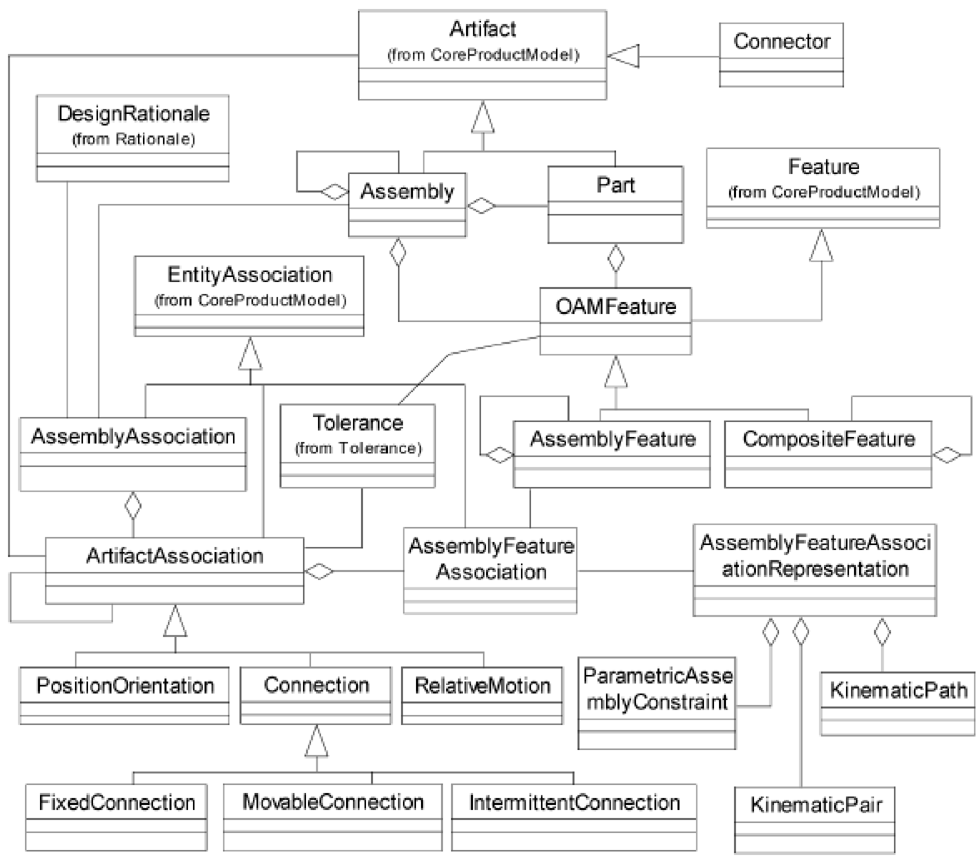


Figure A.5: Main Schema of Open Assembly Model

The class **AssemblyFeatureAssociationRepresentation** represents the assembly relationship between two or more assembly features**.** This class is an aggregation of parametric assembly constraints, a kinematic pair, and/or a relative motion between assembly features. P**arametricAssemblyConstraint** specifies explicit geometric constraints between artifacts of an assembled product, intended to control the position and orientation of artifacts in an assembly. Parametric assembly constraints are defined in ISO 10303-108). This class is further specialized into specific types: **Parallel**, **ParallelWithDimension**, **SurfaceDistanceWithDimension**, **AngleWithDimension**, **Perpendicular**, **Incidence**, **Coaxial**, **Tangent**, and **FixedComponent.**

KinematicPair defines the kinematic constraints between two adjacent artifacts (links) at a joint. The kinematic structure schema in ISO 10303-105 defines the kinematic structure of a mechanical product in terms of links, pairs, and joints. The kinematic pair represents the geometric aspects of the kinematic constraints of motion between two assembled components. **KinematicPath** represents the relative motion between artifacts. The kinematic motion schema in ISO 10303-105 defines kinematic motion. It is also used to represent the relative motion between artifacts.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 A.6.

DimensionalTolerance typically controls the variability of linear dimensions that describe location, size, and angle; it is also known as tolerancing of perfect form. This is included to accommodate the ISO 1101 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 a part surface; feature attributes include size (for certain features), position (certain features), form (flatness, cylindricity, etc.), and relationship (e.g. perpendicular-to). The class **GeometricTolerance** is further specialized into the following: (1) **FormTolerance**; (2) **ProfileTolerance**; (3) **RunoutTolerance**; (4) **OrientationTolerance**; and (5) **LocationTolerance**.

Datum is a theoretically exact or a simulated piece of geometry, such as a point, line, or plane, from which a tolerance is referenced. **DatumFeature** is a physical feature that is applied to establish a datum. **FeatureOfSize** is a feature that is associated with a size dimension, such as the diameter of a spherical or cylindrical surface or the distance between two parallel planes. **StatisticalControl** is a specification that incorporates statistical process controls on the toleranced feature in manufacturing.

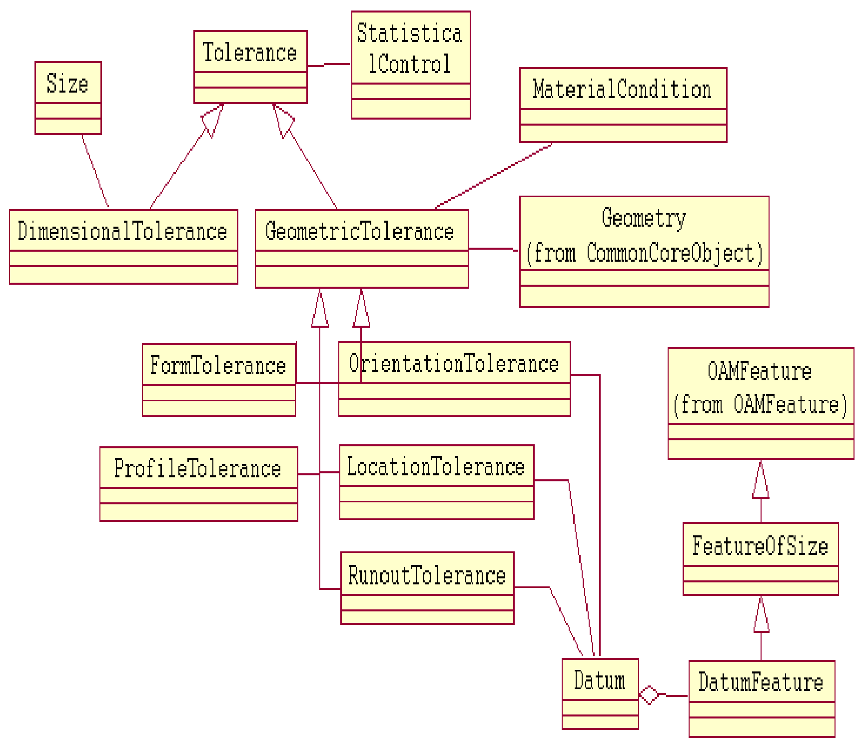


Figure A.6: Tolerance Model

Appendix – 2 Screw Theory Representation and Feature Toolkit

A-2.1 Screw Theory Representations of Assembly Associations (Whitney, 2004)

A screw is a method of demonstrating the motions of a rigid body or the forces and moments acting on it. Screws that characterize motions are called twists or twist matrices, whereas the screws that characterize forces are called wrenches or wrench matrices. A twist or wrench matrix consists of six columns and one to six rows, one for each degree of freedom. These matrices can depict a host of part-to-part constraints. We will utilize them to build a toolkit of useful assembly features. Moreover, we will outline and implement algorithms necessary for motion and constraint analysis of assemblies constructed by combining parts and using assembly features (Whitney, 2004).

The general form of a twist is

$T = [\omega_x\ \omega_y\ \omega_z\ v_x\ v_y\ v_z]$ (1)

The wrench matrix fits the twist matrix and it composed of all the forces and moments that the joint is able to resist. A wrench is defined in the following way:

$W = [fx\ fy\ fz\ mx\ my\ mz]$ ....(2)

For the assembly association between Pin and Gear in figure 4.16, the twist matrices are given as;

$T = [\omega_x\ \omega_y\ \omega_z\ v_x\ v_y\ v_z]$ .... (3)

$T = [\omega_x\ \omega_y\ \omega_z\ v_x\ v_y\ v_z]$ .... (4)

The relative position and orientation of the features and artifacts (the local coordinate system [LCS] for artifacts and the feature coordinate system [FCS] for the features) are defined in chapter 3.

After that, the connection (joint) properties of associated artifacts and assembly features are defined by twist matrices (TR) and frames of the assembly features (or links), based on the degree of freedom of the connection, and stored in AssemblyFeature, AssemblyFeatureAssociation, Connection, KinematicPair classes in the OAM model. When many parts are joined this way, one can navigate from part to part by following the transformation frames.

Where, w is rotational velocity, v is linear velocity, subscript 1 is for the first degree of freedom, and x, y, z stand for the directions.

Another matrix used for defining associations in the assembly is the wrench matrix, which is used for constraint analysis with flow of force and moment. A wrench is a screw that describes the resultant force and moment of a force system acting on a rigid body (in Connection and Function classes). Wrenches matrices are used for constraint analysis with flow of force and moment. The constraint matrix is

$$WR = \begin{bmatrix} f_{1x} & f_{1y} & f_{1z} & M_{1x} & M_{1y} & M_{1z} \\ . & . & . & . & . & . \end{bmatrix} \quad \text{.......} (5)$$

In Section, interactions among associations of assembly, function and behavior (with matrices and relevant OAM model classes) in the product information representation and exchange model are discussed and shown in the planetary gearbox example.

A-2.2. Construction of Twist Matrices

There are two main classes of features used in assembly: features associated with the product function and features associated with part making process. The former contain common joints like cylinder in hole, plate on plate, tongue in groove etc., whereas the latter contain surface plates, pillow blocks, V-blocks, locating pins and their concave matches (holes or slots), V-shaped locators and their concave matches (V-shaped notches), etc.(Whitney, 2004).

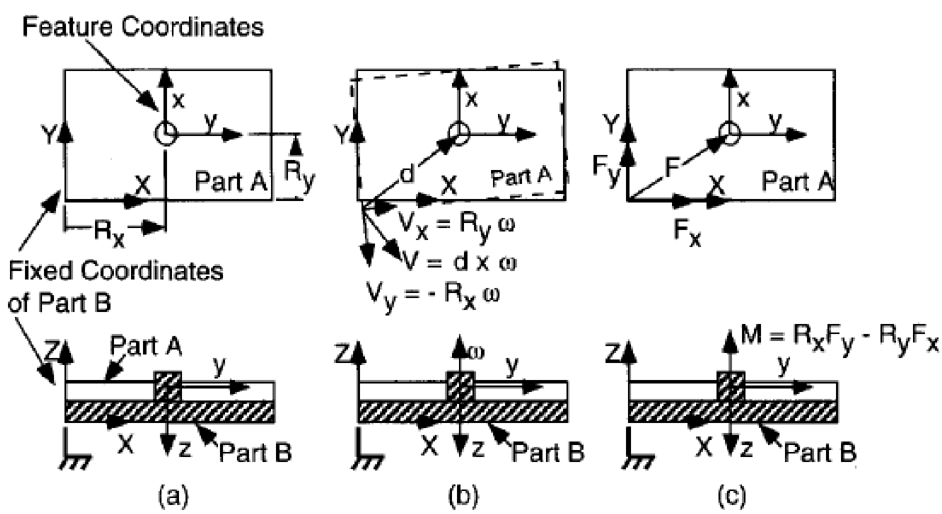


Figure 4.20: Two Flat Plates Joined by a Pin-Hole Joint (Whitney, 2004)

In figure 4.20 (a), definition of coordinates of Part B is regarded as fixed and holds the reference coordinate frame in the lower left corner. The pin is placed at distance R_x in the direction of x and R_y in the direction of y from the origin of Part B's coordinate frame. (b) Part A can freely rotate around the pin's axis and when it rotates at the angular rate &>, a point on Part A overlaps with the origin of Part B coordinate frame and translates at velocity V, that is comprised of x and y components, V_x and V_y, respectively. Part A is limited to only this type of motion. (c) Part A's equilibrium can be maintained by applying external forces and moments, such as one in the plane of the part applied straight to the pin center, as shown here. It corresponds to the separate forces F_x and F_y plus the moment $M = R_xF_y - R_yF_x$. Other forces and moments that can be resisted (not shown) are F_z, M_x, and M_y.

A-2.3. Motion and Constraint Analysis of Assembly Associations

Motion limit analysis is important for behavioral model and tracing of problems in the system/assembly.

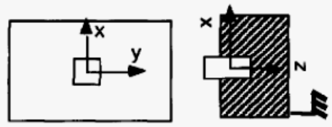
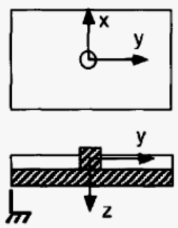
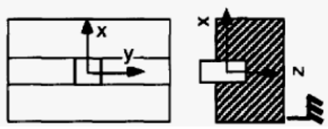
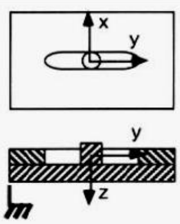
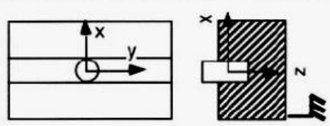
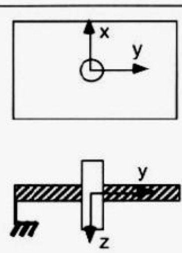
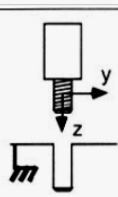
$$TR = \begin{bmatrix} \omega_{1x} & \omega_{1y} & \omega_{1z} & \upsilon_{1x} & \upsilon_{1y} & \upsilon_{1z} \\ \omega_{2x} & \omega_{2y} & \omega_{2z} & \upsilon_{2x} & \upsilon_{2y} & \upsilon_{2z} \\ \omega_{3x} & \dots & & & & \end{bmatrix}$$

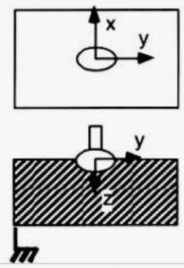
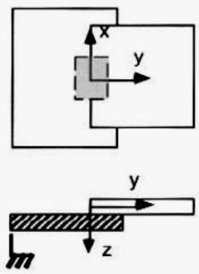
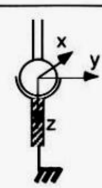
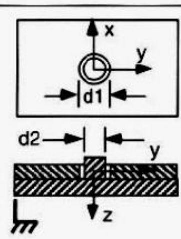
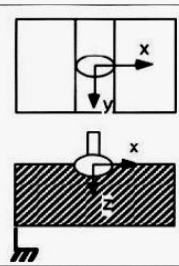
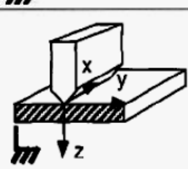
Where, w is rotational velocity, v is linear velocity, subscript 1 is for the first degree of freedom, and x, y, z stand for the directions.

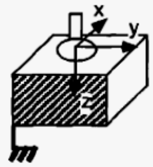
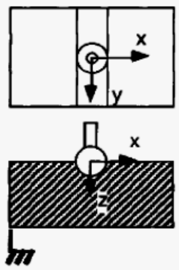
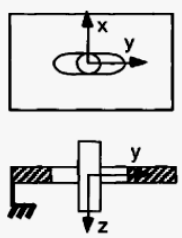
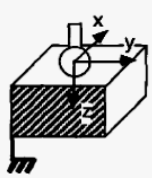
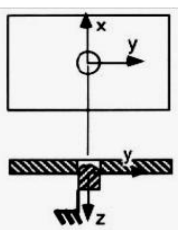
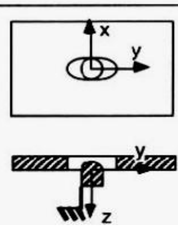
Another matrix used for defining associations in the assembly is the wrench matrix, which is used for constraint analysis with flow of force and moment. A wrench is a screw that describes the resultant force and moment of a force system acting on a rigid body (in Connection and Function classes). Wrenches matrices are used for constraint analysis with flow of force and moment. The constraint matrix is

$$WR = \begin{bmatrix} f_{1x} & f_{1y} & f_{1z} & M_{1x} & M_{1y} & M_{1z} \\ . & . & . & . & . & . \end{bmatrix}$$

A-2.4. Feature Toolkit: Twist and Wrench Matrices for Assembly Constraints (Whitney, 2004)

Toolkit Feature Number and Name	Sketch	Twist Matrix	Remarks
1—Prismatic pin in prismatic hole		$T_1 = [0\ 0\ 0\ 0\ 0\ 0]$	In general, twist matrix entries are in part coordinates, but entities like "ω_x" are in feature coordinates.
2—Pin on plate in hole		$T_2 = [\omega\ \ v]$ where $\omega = (R\omega_z)^T$, $v = r \times \omega$, $r = d^T$	Twist matrix if top plate is very thin: $T_{2'} = \begin{bmatrix} \omega_1 & v_1 \\ \omega_2 & v_2 \\ \omega_3 & v_3 \end{bmatrix}$ where $\omega_1 = (R\omega_z)^T, \omega_2 = (R\omega_x)^T$, and $\omega_3 = (R\omega_y)^T$ and $v_i = r \times \omega_i, i = 1, 2, 3$
3—Prismatic pin and prismatic slot		$T_3 = [0\ \ v]$ where $v = (Rk_y)^T$ and $0 = (0, 0, 0)$	
4—Pin on plate in slot		$T_4 = \begin{bmatrix} \omega_1 & v_1 \\ \omega_2 & v_2 \\ 0 & v_3 \end{bmatrix}$ where $\omega_1 = (R\omega_x)^T$ and $\omega_2 = (R\omega_z)^T$, $v_i = r \times \omega_i, i = 1, 2$, and $v_3 = (Rk_y)^T$	One rotation permits the plate to rotate in the XY plane. The other permits the plate to rotate about the X axis of the pin. If the plate is thin, we can add a third rotation: $T_{4'} = \begin{bmatrix} \omega_1 & v_1 \\ \omega_2 & v_2 \\ 0 & v_3 \\ \omega_4 & v_4 \end{bmatrix}$ where $\omega_4 = (R\omega_y)^T$
5—Round pin in prismatic slot		Same as feature 4	
6—Round pin in hole		$T_6 = \begin{bmatrix} \omega & v_1 \\ 0 & v_2 \end{bmatrix}$ where $\omega = (R\omega_z)^T$ and $v_2 = (Rk_z)^T$	Compared to feature 2, this feature provides a pivot but does not include planar support along the z axis.
7—Threaded joint		$T_7 = [\omega\ \ v]$ where $v = r \times \omega + p\omega$	p is the pitch of the threads

Toolkit Feature Number and Name	Sketch	Twist Matrix	Remarks
8—Elliptical ball and socket		$T_8 = \begin{bmatrix} \omega_1 & v_1 \\ \omega_2 & v_2 \end{bmatrix}$	
9—Plate–plate lap joint		$T_9 = \begin{bmatrix} \omega & 0 \\ 0 & v_1 \\ 0 & v_2 \end{bmatrix}$ where $\omega = (R\omega_z)^T$, v_1 = any vector perpendicular to ω, and v_2 is perpendicular to both v_1 and ω.	The 0 to the right of ω indicates that there is no fixed rotation axis in this case, so there is no definite velocity arising from ω. The lightly shaded area in the sketch represents the allowable location of the coordinate frame for the lapping part.
10—Spherical joint		$T_{10} = \begin{bmatrix} \omega_1 & v_1 \\ \omega_2 & v_2 \\ \omega_3 & v_3 \end{bmatrix}$ where $\omega_1 = (R\omega_x)^T$, $\omega_2 = (R\omega_y)^T$, $\omega_3 = (R\omega_z)^T$, $v_1 = r \times \omega_1$, $v_2 = r \times \omega_2$, $v_3 = r \times \omega_3$	
11—Pin in oversize hole		$T_{11} = \begin{bmatrix} \omega & 0 \\ 0 & v_1 \\ 0 & v_2 \end{bmatrix}$ where $\omega = (R\omega_z)^T$, v_1 = any vector perpendicular to ω, and v_2 is perpendicular to both v_1 and ω.	The 0 to the right of ω indicates that there is no fixed rotation axis in this case, so there is no definite velocity arising from ω. We can add two rows to express the fact that the pin can rock in the clearance about the feature's x and y axes. These extra motions are also possible if the upper plate is very thin.
12—Elliptical ball in cylindrical groove		$T_{12} = \begin{bmatrix} \omega_1 & v_1 \\ \omega_2 & v_2 \\ 0 & v_3 \end{bmatrix}$ where v_1 and v_2 are defined as usual, and $v_3 = (Rk)^T$.	
13—Edge on plane		$T_{13} = \begin{bmatrix} \omega_1 & v_1 \\ \omega_2 & 0 \\ 0 & v_2 \\ 0 & v_3 \end{bmatrix}$ where $\omega_1 = (R\omega_x)^T$, $v_2 = (Rk_x)^T$, and $v_3 = (Rk_y)^T$.	

Toolkit Feature Number and Name	Sketch	Twist Matrix	Remarks
14—Ellipsoid on plane		$T_{14} = \begin{bmatrix} \omega_x & v_1 \\ \omega_y & v_2 \\ \omega_z & v_3 \\ 0 & v_4 \\ 0 & v_5 \end{bmatrix}$ where $v_4 = (Rk_x)^T$ and $v_5 = (Rk_y)^T$.	
15—Sphere in cylindrical trough		$T_{15} = \begin{bmatrix} \omega_1 & v_1 \\ \omega_2 & v_2 \\ \omega_3 & v_3 \\ 0 & v_4 \end{bmatrix}$ where $\omega_1 = (A\omega_x)^T$ $\omega_2 = (A\omega_y)^T$ $\omega_3 = (A\omega_z)^T$ $v_1 = r \times \omega_1$ $v_2 = r \times \omega_2$ $v_3 = r \times \omega_3$ and $v_4 = (Ak_y)^T$.	
16—Pin in slot		$T_{16} = \begin{bmatrix} \omega_1 & v_1 \\ \omega_2 & v_2 \\ 0 & v_3 \\ 0 & v_4 \end{bmatrix}$ where $\omega_1 = (R\omega_z)^T$, $\omega_2 = (R\omega_x)^T$, $v_3 = (Rk_z)^T$, and $v_4 = (Rk_y)^T$.	If we want to capture the case where the upper plate is very thin and the pin can rock about the its y axis, we can add a row to the twist matrix to obtain $T_{16} = \begin{bmatrix} \omega_1 & v_1 \\ \omega_2 & v_2 \\ 0 & v_3 \\ 0 & v_4 \\ \omega_5 & v_5 \end{bmatrix}$ where $\omega_5 = (R\omega_y)^T$.
17—Sphere on plane		$T_{17} = \begin{bmatrix} \omega_1 & v_1 \\ \omega_2 & v_2 \\ \omega_3 & 0 \\ 0 & v_3 \\ 0 & v_4 \end{bmatrix}$ where the rotations are defined as in feature 15, $v_3 = (Rk_x)^T$, and $v_4 = (Rk_y)^T$.	
18—Hemispherical pin in hole		$T_{18} = \begin{bmatrix} \omega_1 & v_1 \\ \omega_2 & v_2 \\ \omega_3 & v_3 \\ 0 & v_4 \end{bmatrix}$	Compared to feature 2, this feature provides a pivot axis about z, permits motion along z, and permits rotation about x and y.
19—Hemispherical pin in slot		$T_{19} = \begin{bmatrix} \omega_1 & v_1 \\ \omega_2 & v_2 \\ \omega_3 & v_3 \\ 0 & v_4 \\ 0 & v_5 \end{bmatrix}$	

Appendix – 3. Reconciled Definitions for Flow and Function Taxonomy

A-3.1. Flow Definitions (Hirtz et. al. ,2002)

1) **Material**

a) **Human**. All or part of a person who crosses the device boundary. Example: Most coffee makers require the flow of a *human hand* to actuate (or start) the electricity and thus heat the water.

b) **Gas**. Any collection of molecules characterized by random motion and he absence of bonds between the molecules. Example: An oscillating fan moves air by rotating blades.The air is transformed as *gas* flow.

c) **Liquid**. A readily flowing fluid, specifically having its molecules moving freely withrespect to each other, but because of cohesive forces, not expanding indefinitely.Example: The flow of water through a coffee maker is a *liquid*.

d) **Solid**. Any object with mass having a definite, firm shape. Example: The flow ofsandpaper into a hand sander is transformed into a *solid* entering the sander.

i) **Object**. Material that can be seen or touched that occupies space. Example: The boxof scrap paper for recycling is represented as the flow *object*.

ii) **Particulate**. Substance containing minute separate particles. Example: Granularsugar and powdered paint are *particulates*.

iii) **Composite**. Solid material composed of two or more substances having differentphysical characteristics and in which each substance retains its identity whilecontributing desirable properties to the whole unit. Any class of high-strength,lightweight engineering materials consisting of various combinations of alloys,plastics, and ceramics. Example: Materials such as wood, fiberglass combined with metals, ceramics, glasses, or polymers together are considered a *composite*. Kevlar cloth combined with paper honeycomb by means of a resin is considered a *composite*.

e) **Plasma.** A collection of charged particles that is electrically neutral exhibiting some properties of a gas, but differing from a gas in being a good conductor of electricity and in being affected by a magnetic field. Example: Plasma cutting focuses an intense beam of ionized air, known as *plasma*, produced by an electric arc, which melts the material to be cut.

f) **Mixture**. A substance containing two or more components which are not in fixed proportions, do not lose their individual characteristics and can be separated by physical means. Example: Expected precipitation for this evening is a *mixture* of rain, sleet, and snow.

i) **Liquid-liquid**. A readily flowing combination of two or more fluids, specifically having its molecules moving freely with respect to each other, but because of cohesive forces, not expanding indefinitely. Example: Machine oil and gasoline is a common *liquid-liquid* mixture used in yard maintenance machines.

ii) **Gas-gas**. A collection of molecules containing two or more components, which are characterized by random motion and the absence of bonds between the molecules. Example: The mixture of argon and carbon dioxide, a *gas-gas* flow, is commonly used in welding.

iii) **Solid-solid**. A combination of two or more objects with mass having definite, firm shape. Example: Pebbles, sand, gravel, and slag can be used to form concrete, mortar, or plaster. After it cures, concrete is a *solid-solid.*

iv) **Solid-Liquid**. A combination of two or more components containing at least one solid and one liquid. Example: Iced Tea is a *solid-liquid* mixture of ice (solid), water (liquid), and tea grounds (solid).

v) **Solid-Gas**. A combination of two or more components containing at least one solid and one gas. Example: Fog is a *solid-gas* mixture of frozen ice particles (solid) in air (gas).

vi) **Liquid-Gas**. A combination of two or more components containing at least one liquid and one gas. Example: Carbonated drinks are *liquid-gas* mixtures of flavored syrup (liquid), purified water (liquid), and carbon dioxide (gas).

vii) **Solid-Liquid-Gas**. A combination or three or more components containing at least one each of a solid, liquid, and gas. Example: In a cup of soda and ice cubes, the cup contains the *solid-liquid-gas* flow.

viii) **Colloidal**. A solid, liquid, or gaseous substance made up of very small, insoluble non-diffusible particles that remain in suspension in a surrounding solid, liquid, or gaseous medium of a different matter. Example: Aerosols, smoke, and mist can all be considered *colloids*. Mist is a combination of very fine water droplets suspended in air.

2) Energy

a) **Generic Complements**.

i) **Effort**. Any component of energy used to accomplish an intended purpose.

ii) **Flow**. Any component of energy causing the intended object to move or run freely.

b) **Human**. Work performed by a person on a device. Example: An automobile requires the flow of *human energy* to steer and accelerate the vehicle.

i) **Force**. Human effort that is input to the system without regard for the required motion. Example: *Human force* is needed to actuate the trigger of a toy gun.

ii) **Velocity**. Activity requiring movement of all or part of the body through a prescribed path. Example: The track pad on a laptop computer receives the flow of *human velocity* to control the cursor.

c) **Acoustic**. Work performed in the production and transmission of sound. Example: The motor of a power drill generates the flow of *acoustic energy* in addition to the torque.

i) **Pressure**. The pressure field of the sound waves. Example: A condenser microphone has a diaphragm, which vibrates in response to *acoustic pressure*. This vibration changes the capacitance of the diaphragm, thus superimposing an alternating voltage on the direct voltage applied to the circuit.

ii) **Particle velocity**. The speed at which sound waves travel through a conducting medium. Example: Sonar devices rely on the flow of *acoustic particle velocity* to determine the range of an object.

d) **Biological**. Work produced by or connected with plants or animals. Example: In poultry houses, grain is fed to chickens, which is then converted into *biological energy*.

i) **Pressure**. The pressure field exerted by a compressed biological fluid. Example: The high concentration of sugars and salts inside a cell causes the entry, via osmosis, of water into the vacuole, which in turn expands the vacuole and generates a hydrostatic *biological pressure*, called turgor, that presses the cell membrane against the cell wall. Turgor is the cause of rigidity in living plant tissue.

ii) **Volumetric flow**. The kinetic energy of molecules in a biological fluid flow. Example: Increased metabolic activity of tissues such as muscles or the intestine automatically induces increased *volumetric flow* of blood through the dilated vessels.

e) **Chemical**. Work resulting from the reactions by which substances are produced from or converted into other substances. Example: A battery converts the flow of *chemical energy* into electrical energy.

i) **Affinity**. The force with which atoms are held together in chemical bonds. Affinity is proportional to the chemical potential of a compound's constituent species. Example: An internal combustion engine transforms the chemical *affinity* of the gas into a mechanical force.

ii) **Reaction rate**. The speed or velocity at which chemical reactants produce products. Reaction rate is proportional to the mole rate of the constituent species. Example: Special coatings on automobile panels stop the *chemical reaction rate* of the metal with the environment.

f) **Electrical**. Work resulting from the flow of electrons from a negative to a positive source. Example: A power belt sander imports a flow of *electrical energy* (electricity, for convenience) from a wall outlet and transforms it into a rotation.

i) **Electromotive force**. Potential difference across the positive and negative sources. Example: Household electrical receptacles provide a flow of *electromotive force* of approximately 110 V.

ii) **Current**. The flow or rate of flow of electric charge in a conductor or medium between two points having a difference in potential. Example: Circuit breakers trip when the *current* exceeds a specified limit.

g) **Electromagnetic**. Energy that is propagated through free space or through a material medium in the form of electromagnetic waves (Britannica Online, 1997). It has both wave and particle-like properties. Example: Solar panels convert the flow *electromagnetic energy* into electricity.

i) **Generic Complements**.

(1) **Effort**. Any component of electromagnetic energy used to accomplish an intended purpose.

(2) **Flow**. Any component of electromagnetic energy causing the intended object to move or run freely.

ii) **Optical**. Work associated with the nature and properties of light and vision. Also, a special case of solar energy (see solar). Example: A car visor refines the flow of *optical energy* that its passengers receive.

(1) **Intensity**. The amount of optical energy per unit area. Example: Tinted windows reduce the *optical intensity* of the entering light.

(2) **Velocity**. The speed of light in its conducting medium. Example: NASA developed and tested a trajectory control sensor (TCS) for the space shuttle to calculate the distance between the payload bay and a satellite. It relied

on the constancy of the *optical velocity* flow to calculate distance from time of flight measurements of a reflected laser.

iii) **Solar**. Work produced by or coming from the sun. Example: Solar panels collect the flow of *solar energy* and transform it into electricity.

(1) **Intensity**. The amount of solar energy per unit area. Example: A cloudy day reduces the *solar intensity* available to solar panels for conversion to electricity.

(2) **Velocity**. The speed of light in free space. Example: Unlike most energy flows,*solar velocity* is a well-known constant.

h) **Hydraulic**. Work that results from the movement and force of a liquid, including hydrostatic forces. Example: Hydroelectric dams generate electricity by harnessing the *hydraulic energy* in the water that passes through the turbines.

i) **Pressure**. The pressure field exerted by a compressed liquid. Example: A hydraulic jack uses the flow *hydraulic pressure* to lift heavy objects.

ii) **Volumetric flow**. The movement of fluid molecules. Example: A water meter measures the *volumetric flow* of water without a significant pressure drop in the line.

i) **Magnetic**. Work resulting from materials that have the property of attracting other like materials, whether that quality is naturally occurring or electrically induced. Example: The *magnetic energy* of a magnetic lock is the flow that keeps it secured to the iron based structure.

i) **Magnetomotive force**. The driving force which sets up the magnetic flux inside of a core. Magnetomotive force is directly proportional to the current in the coil surrounding the core. Example: In a magnetic door lock, a change in *magnetomotive force* (brought about by a change in electrical current) allows the lock to disengage and the door to open.

ii) **Magnetic flux rate**. Flux is the magnetic displacement variable in a core induced by the flow of current through a coil. The magnetic flow variable is the time rate of change of the flux. The voltage across a magnetic coil is directly proportional to the time rate of change of magnetic flux. Example: A magnetic relay is a transducer that senses the time rate of change of *magnetic flux* when the relay arm moves.

j) **Mechanical**. Energy associated with the moving parts of a machine or the strain energy associated with a loading state of an object. Example: An elevator converts electrical or hydraulic energy into mechanical energy.

i) **Generic Complements**.

(1) **Effort**. Any component of mechanical energy used to accomplish an intended purpose.

(2) **Flow**. Any component of mechanical energy causing the intended object to move or run freely.

ii) **Rotational energy**. Energy that results from a rotation or a virtual rotation. Example: Customers are primarily concerned with the flow of *rotational energy* from a power screwdriver.

(1) **Torque**. Pertaining to the moment that produces or tends to produce rotation. Example: In a power screwdriver, electricity is converted into rotational energy. The more specific flow is *torque*, based on the primary customer need to insert screws easily, not quickly.

(2) **Angular velocity**. Pertaining to the orientation or the magnitude of the time rate of change of angular position about a specified axis. Example: A centrifuge is used to separate out liquids of different densities from a mixture. The primary flow it produces is that of *angular velocity*, since the rate of rotation about an axis is the main concern.

iii) **Translational energy**. Energy flow generated or required by a translation or a virtual translation. Example: A child's toy, such as a projectile launcher, transmits *translational energy* to the projectile to propel it away.

(1) **Force**. The action that produces or attempts to produce a translation. Example: In a tensile testing machine, the primary flow of interest is that of a *force* which produces a stress in the test specimen.

(2) **Linear velocity**. Motion that can be described by three component directions. Example: An elevator car uses the flow of *linear velocity* to move between floors.

k) **Pneumatic**. Work resulting from a compressed gas flow or pressure source. Example: A BB gun relies on the flow of *pneumatic energy* (from compressed air) to propel the projectile (BB).

i) **Pressure**. The pressure field exerted by a compressed gas. Example: Certain cylinders rely on the flow of *pneumatic pressure* to move a piston or support a force.

ii) **Mass flow**. The kinetic energy of molecules in a gas flow. Example: The *mass flow* of air is the flow that transmits the thermal energy of a hair dryer to damp hair.

l) **Radioactive (Nuclear)**. Work resulting from or produced by particles or rays, such as alpha, beta and gamma rays, by the spontaneous disintegration of atomic nuclei. Example: Nuclear reactors produce a flow of *radioactive energy* which heats water into steam and then drives electricity generating turbines.

i) **Intensity**. The amount of radioactive particles per unit area. Example: Concrete is an effective radioactive shielding material, reducing the *radioactive intensity* in proportion to its thickness.

ii) **Decay rat**e. The rate of emission of radioactive particles from a substance. Example: The *decay rate* of carbon provides a method to date pre-historic objects.

m) **Thermal**. A form of energy that is transferred between bodies as a result of their temperature difference. Example: A coffee maker converts the flow of electricity into the flow of *thermal energy*, which it transmits to the water. Note: A pseudo bond graph approach is used here. The true effort and flow variables are temperature and the time rate of change of entropy. However, a more practical pseudo-flow of heat rate is chosen here.

i) **Temperature**. The degree of heat of a body. Example: A coffee maker brings the *temperature* of the water to boiling in order to siphon the water from the holding tank to the filter basket.

ii) **Heat rate**. (Note: this is a pseudo-flow) The time rate of change of heat energy of a body. Example: Fins on a motor casing increase the flow *heat rate* from the motor by conduction (through the fin), convection (to the air) and radiation (to the environment).

3) **Signal**

a) **Status**. A condition of some system, as in information about the state of the system. Example: Automobiles often measure the engine water temperature and send a *status signal* to the driver via a temperature gage.

i) **Auditory**. A condition of some system as displayed by a sound. Example: Pilots receive an *auditory signal*, often the words "pull up," when their aircraft reaches a dangerously low altitude.

ii) **Olfactory**. A condition of some system as related by the sense of smell or particulate count. Example: Carbon monoxide detectors receive an *olfactory signa*l from the environment and monitor it for high levels of CO.

iii) **Tactile**. A condition of some system as perceived by touch or direct contact. Example: A pager delivers a *tactile signal* to its user through vibration.

iv) **Taste**. A condition of some dissolved substance as perceived by the sense of taste. Example: In an electric wok, the *taste signal* from the human chef is used to determine when to turn off the wok.

v) **Visual**. A condition of some system as displayed by some image. Example: A power screwdriver provides a *visual signal* of its direction through the display of arrows on the switch.

b) **Control**. A command sent to an instrument or apparatus to regulate a mechanism. Example: An airplane pilot sends a *control signal* to the elevators through movement of the yoke. The yoke movement is transformed into an electrical signal, sent through wiring to the elevator, and then transformed back into a physical elevator deflection.

i) **Analog**. A control signal sent by direct, continuous, measurable, variable physical quantities. Example: Turning the volume knob on a radio sends an *analog signal* to increase or decrease the sound level.

ii) **Discrete**. A control signal sent by separate, distinct, unrelated or discontinuous quantities. Example: A computer sends *discrete signals* to the hard disk controller during read/write operations.

A-3.2. Function Definitions (Hirtz et. al. ,2002)

Note that certain functions are limited to operate on certain types of flows. This restriction is typically given in the function definition and applies to all functions at sub-levels of the given function.

1) **Branch**. To cause a flow (material, energy, signal) to no longer be joined or mixed.

a) Separate. To isolate a flow (material, energy, signal) into distinct components. The separated components are distinct from the flow before separation, as well as each other. Example: A glass prism *separates* light into different wavelength components to produce a rainbow.

i) **Divide**. To separate a flow. Example: A vending machine divides the solid form of coins into appropriate denominations.

ii) **Extract**. To draw, or forcibly pull out, a flow. Example: A vacuum cleaner *extracts* debris from the imported mixture and exports clean air to the environment.

iii) **Remove**. To take away a part of a flow from its prefixed place. Example: A sander *removes* small pieces of the wood surface to smooth the wood.

b) **Distribute**. To cause a flow (material, energy, signal) to break up. The individual bits are similar to each other and the undistributed flow. Example: An atomizer *distributes* (or sprays) hair-styling liquids over the head to hold the hair in the desired style.

2) **Channel**. To cause a flow (material, energy, signal) to move from one location to another location.

a) **Import**. To bring in a flow (material, energy, signal) from outside the system boundary. Example: A physical opening at the top of a blender pitcher *imports* a solid (food) into the system. Also, a handle on the blender pitcher imports a human hand.

b) **Export**. To send a flow (material, energy, signal) outside the system boundary. Example: Pouring blended food out of a standard blender pitcher *exports* liquid from the system. The opening at the top of the blender is a solution to the export subfunction.

c) **Transfer**. To shift, or convey, a flow (material, energy, signal) from one place to another.

i) **Transport**. To move a material from one place to another. Example: A coffee maker *transports* liquid (water) from its reservoir through its heating chamber and then to the filter basket.

ii) **Transmit**. To move an energy from one place to another. Example: In a hand held power sander, the housing of the sander *transmits* human force to the object being sanded.

d) **Guide**. To direct the course of a flow (material, energy, signal) along a specific path. Example: A domestic HVAC system *guides* gas (air) around the house to the correct locations via a set of ducts.

i) **Translate**. To fix the movement of a flow by a device into one linear direction. Example: In an assembly line, a conveyor belt *translates* partially completed products from one assembly station to another.

ii) **Rotate**. To fix the movement of a flow by a device around one axis. Example: A computer disk drive *rotates* the magnetic disks around an axis so that the head can read data.

iii) **Allow degree of freedom (DOF)**. To control the movement of a flow by a force external to the device into one or more directions. Example: To provide easy trunk access and close appropriately, trunk lids need to move along a specific degree of freedom. A four bar linkage *allows a rotational DOF* for the trunk lid.

3) **Connect**. To bring two or more flows (material, energy, signal) together.

a) **Couple**. To join or bring together flows (material, energy, signal) such that the members are still distinguishable from each other. Example: A standard pencil couples an eraser and a writing shaft. The coupling is performed using a metal sleeve that is crimped to the eraser and the shaft.

i) **Join**. To couple flows together in a predetermined manner. Example: A ratchet *joins* a socket on its square shaft interface.

ii) **Link**. To couple flows together by means of an intermediary flow. Example: A turnbuckle *links* two ends of a steering cable together.

b) **Mix**. To combine two flows (material, energy, signal) into a single, uniform homogeneous mass. Example: A shaker *mixes* a paint base and its dyes to form a homogeneous liquid.

4) **Control Magnitude**. To alter or govern the size or amplitude of a flow (material, energy, signal).

a) **Actuate**. To commence the flow of energy, signal, or material in response to an imported control signal. Example: A circuit switch *actuates* the flow of electrical energy and turns on a light bulb.

b) **Regulate**. To adjust the flow of energy, signal, or material in response to a control signal, such as a characteristic of a flow. Example: Turning the valves *regulates* the flow rate of the liquid flowing from a faucet.

i) **Increase**. To enlarge a flow in response to a control signal. Example: Opening the valve of a faucet further *increases* the flow of water.

ii) **Decrease**. To reduce a flow in response to a control signal. Example: Closing the value further *decreases* the flow of propane to the gas grill.

c) **Change**. To adjust the flow of energy, signal, or material in a predetermined and fixed manner. Example: In a hand held drill, a variable resistor *changes* the electrical energy flow to the motor thus changing the speed the drill turns.

i) **Increment**. To enlarge a flow in a predetermined and fixed manner. Example: A magnifying glass *increments* he visual signal (i.e. the print) from a paper document.

ii) **Decrement**. To reduce a flow in a predetermined and fixed manner. Example: The gear train of a power screwdriver *decrements* the flow of rotational energy.

iii) **Shape**. To mold or form a flow. Example: In the auto industry, large presses *shape* sheet metal into contoured surfaces that become fenders, hoods and trunks.

iv) **Condition**. To render a flow appropriate for the desired use. Example: To prevent damage to electrical equipment, a surge protector *conditions* electrical energy by excluding spikes and noise (usually through capacitors) from the energy path.

d) **Stop**. To cease, or prevent, the transfer of a flow (material, energy, signal). Example: A reflective coating on a window *stops* the transmission of UV radiation through a window.

i) **Prevent**. To keep a flow from happening. Example: A submerged gate on a dam wall *prevents* water from flowing to the other side.

ii) **Inhibit**. To significantly restrain a flow, though a portion of the flow continues to be transferred. Example: The structures of space vehicles *inhibits* the flow of radiation to protect crew and cargo.

5) **Convert**. To change from one form of a flow (material, energy, signal) to another. For completeness, any type of flow conversion is valid. In practice, conversions such as convert electricity to torque will be more common than convert solid to optical energy. Example: An electrical motor *converts* electricity to rotational energy.

6) **Provision**. To accumulate or provide a material or energy flow.

a) **Store**. To accumulate a flow. Example: A DC electrical battery *stores* the energy in a flashlight.

i) **Contain**. To keep a flow within limits. Example: A vacuum bag *contains* debris vacuumed from a house.

ii) **Collect**. To bring a flow together into one place. Example: Solar panels *collect* ultraviolet sun rays to power small mechanisms.

b) **Supply**. To provide a flow from storage. Example: In a flashlight, the battery *supplies* energy to the bulb.

7) **Signal**. To provide information on a material, energy or signal flow as an output signal flow. The information providing flow passes through the function unchanged.

a) **Sense**. To perceive, or become aware, of a flow. Example: An audiocassette machine *senses* if the end of the tape has been reached.

i) **Detect**. To discover information about a flow. Example: A gauge on the top of a gas cylinder *detects* proper pressure ranges.

ii) Measure. To determine the magnitude of a flow. Example: An analog thermostat *measures* temperature through a bimetallic strip.

b) **Indicate**. To make something known to the user about a flow. Example: A small window in the water container of a coffee maker *indicates* the level of water in the machine.

i) **Track**. To observe and record data from a flow. Example: By *tracking* the performance of batteries, the low efficiency point can be determined.

ii) **Display**. To reveal something about a flow to the mind or eye. Example: The xyzcoordinate display on a vertical milling machine *displays* the precise location of the cutting tool.

c) **Process**. To submit information to a particular treatment or method having a set number of operations or steps. Example: A computer *processes* a login request signal before allowing a user access to its facilities.

8) **Support**. To firmly fix a material into a defined location, or secure an energy or signal into a specific course.

a) **Stabilize**. To prevent a flow from changing course or location. Example: On a typical canister vacuum, the center of gravity is placed at a low elevation to *stabilize* the vacuum when it is pulled by the hose.

b) **Secure**. To firmly fix a flow path. Example: On a bicycling glove, a Velcro strap *secures* the human hand in the correct place.

c) **Position**. To place a flow (material, energy, signal) into a specific location or orientation. Example: The coin slot on a soda machine *positions* the coin to begin the coin evaluation and transportation procedure.

References

Adams, J. D., 1998, “Feature-based Analysis of Selective Limited Motion in Assemblies”, MS thesis, Mechanical Engineering Department, Massachusetts Institute of Technology.

Al-Hakim L, Kusiak A, Mathew J., 2000, “A graph-theoretic approach to conceptual design with functional perspectives”, Computer-Aided Design 2000, 32 pp. 867-875.

Baysal, M.M., Roy, U., Sudarsan, R., Sriram, R.D., and Lyons, K.W, 2005, “Product Information Exchange using Open Assembly Model: Issues Related to Representation of Geometric Information”, Proceedings of the 2005 ASME IMECE, November 5 -11, 2005 Orlando, FL Paper No: IMECE2005-82812.

Baysal, M.M., Roy, U., Sudarsan, R., Sriram, R.D., and Lyons, K.W. (2004) “The Open Assembly Model for the Exchange of Assembly and Tolerance Information: Overview and Example”, Proceedings of The 2004 ASME IDETC, Salt Lake City, Utah.

Bertsche, Bernd, 2008, “Reliability in Automotive and Mechanical Engineering: Determination of Component and System Reliability”, 2008 Springer-Verlag Berlin Heidelberg (ISBN: 978-3-540-33969-4)

Bharadwaj B., 1998, “The Role of Product Specifications in Integrated Product And Process Design.” Ph.D. dissertation, Syracuse University, United States -- New York. (Publication No. AAT 9919240).

Bock, C. E., Zha, X., Suh, H. and Lee, J., 2010, "Ontological Product Modeling for Collaborative Product Design", NISTIR 7643, Gaithersburg, MD 20899.

Bock, C., 2005, "Systems Engineering in the Product Lifecycle", International Journal of Product Development, Vol. 2, No. 1/2, pp. 123-137, 2005.

Bronsvoort WF, and Noort A , 2004, "Multiple-View Feature Modelling For Integral Product Development", Computer-Aided Design 36(10): 929-946.

Brunnermeier, S. B. and Martin, S. A., 1999, "Interoperability Cost Analysis of the U.S. Automotive Supply Chain", Prepared for NIST, RTI Project Number 7007-03, March 1999, Gaithersburg, MD 20899.

Campbell MI, Cagan J, Kotovsky K., 1999, "A-Design: An Agent-Based Approach to Conceptual Design in a Dynamic Environment", Research in Engineering Design 1999 (11):172-92.

Chandrasekaran, B., Goel, A., & Iwasaki, Y., 1993, "Functional Representation as a Basis for Design Rationale". IEEE Computer, 26(1):48-56.

Chang,E.,Li, X., and Schmidt, L.C., 2000, "The Need for Form, Function, and Behavior-based Representations in Design". Lab Reports http://www.enme.umd.edu/DATLab/ Abstracts/ 2000_chang_lab_report.PDF

Fenves, S.J., 2001, "A Core Product Model for Representing Design Information, NISTIR 6736, Gaithersburg, MD 20899, 2001.

Fiorentini, X., Gambino, I., Liang, V., Foufou, S., Sudarsan, R., Mani, M. and Bock, C. E., 2007, "An Ontology for Assembly Representation", NISTIR 7436, National Institute of Standard and Technology, Gaithersburg, MD 20899.

Gallaher, M. P., O'Connor, A. C., Dettbarn, J. L., Jr. and Gilday, L. T., 2004, “Cost Analysis of Inadequate Interoperability in the U.S. Capital Facilities Industry”, NIST GCR 04-867, National Institute of Standard and Technology Advanced Technology Program Information Technology and Electronics Office, Gaithersburg, Maryland 20899.

Gero, J.S ,1990, “Design Prototypes: A Knowledge Representation Schema for Design”, AI Magazine, 11(4), pp. 26–36.

Gorti, S.R. and Sriram, R.D., 1996, “From Symbol to Form: A Framework for Conceptual Design”, Computer-Aided Design, 28(11), pp. 853–870.

Hirtz, J., Stone, R. B., McAdams, D. A., Szykman, S. and Wood, K. L., 2002, "A Functional Basis for Engineering Design: Reconciling and Evolving Previous Efforts" NIST Technical Note 1447, Gaithersburg, MD 20899.

Hu, J. and Peng, Y., 2011, “Development of a Function Oriented Computer Aided Tolerancing (FOCAT) System”, Proceedings of the Institution of Mechanical Engineers, Part B: Journal of Engineering Manufacture, 2011, Vol. 225 pp.1189-1203

ISO 10303-1:1994, “Industrial Automation Systems and Integration -- Product Data Representation and Exchange -- Part 1: Overview and Fundamental Principles”, ISO, Geneva, Switzerland.

ISO 10303-105:1996, “Industrial Automation Systems and Integration -- Product Data Representation and Exchange -- Part 105: Integrated Application Resource: Kinematics”, ISO, Geneva, Switzerland.

ISO 10303-109:2004, “Industrial Automation Systems and Integration -- Product Data Representation and Exchange -- Part 109: Integrated Application Resource: Kinematic and Geometric Constraints for Assembly Models”, ISO, Geneva, Switzerland.

ISO 1101:1983, “Technical Drawings -- Geometrical Tolerancing -- Tolerancing of Form, Orientation, Location and Run-Out -- Generalities, Definitions, Symbols, Indications On Drawings”, International Organization for Standardization, Geneva, Switzerland.

ISO/DIS 10303-108:2003, “Product Data Representation and Exchange: Integrated Application Resource: Parameterization and Constraints for Explicit Geometric Product Models”, International Organization for Standardization (ISO), Geneva, Switzerland.

Iwasaki, Y. and Chandrasekaran, B. 1992. “Design Verification Through Function- and Behavior-Oriented Representation”, Proceedings of the Conference on Artificial Intelligence in Design '92, Gero, J.S., (ed.), Kluwer Academic Publishers, pp. 597-616.

Iwasaki, Y., Vescovi, M, Fikes, R. and Chandrasekaran, B., 1995, “Causal Functional Representation Language with Behavior-Based Semantics”, Applied Artificial Intelligence, 9 (1995), pp. 5–31.

Kim, Kyoung-Yun, Chin, Seongah, Kwon, Ohbyung and Ellis, R. Darin, 2009, "Ontology-Based Modeling and Integration of Morphological Characteristics of Assembly Joints For Network-Based Collaborative Assembly Design", Artificial Intelligence for Engineering Design, Analysis and Manufacturing, 23, pp 71-88.

Kimber, W. E., 1999, “XML Representation Methods for EXPRESS-Driven Data”, NIST GCR 99-781, ISOGEN International Corporation.

Kirschman, C. F., and Fadel, G. M., 1997, “Customer Metrics for the Selection of Generic Forms at the Conceptual Stage of Design,” ASME Design Technical Conferences, Design Theory and Methodology Conference, Sacramento, CA, Aug.1997, Published in Conference Proceedings, DETC97/DTM-3877, pp. 3877.

Kitamura, Y., Kasai, T. and Mizoguchi, R., 2001, “Ontology-based Description of Functional Design Knowledge and its Use in a Functional Way Server”, Proc. of the Pacific Asian Conference on Intelligent Systems 2001, PAIS 2001.

Lombeyda, Santiago and Regli, William C., 1999, "Conceptual Design for Assembly", In Satyandra K. Gupta, editor, ASME Design Engineering Technical Conferences, Fourth Design for Manufacturing Conference, New York, NY, USA, September 12-16 1999. ASME, ASME Press. Las Vegas, NV.

Lubell, J, Peak, R. S., Srinivasan, V., and Waterbury, S. C., 2004, "STEP, XML, and UML: Complementary Technologies", 24th Annual Computers & Information in Engineering Conference, Salt Lake City, Utah, September 2004, CIE-2004-141.

MOKA, 2005 "MOKA: A Framework For Structuring And Representing Engineering Knowledge", http://www.kbe.coventry.ac.uk/moka/miginfo.htm .

Oliver, D.W.; Kelliher, T.P., Keegan, J.G.K., 1997, "Engineering Complex Systems with Models and Objects, McGraw-Hill, 1997

Otto, K. and Wood, K., 2001, "Product Design – Techniques in Reverse Engineering and New Product Development", Prentice Hall, .

Pahl, G., Beitz, W., Feldhusen, J., and Grote, K.-H., 2007, "Engineering Design: A Systematic Approach", (3rd ed.), Springer-Verlag.

Paige, R.F., Olsen, G.K., Kolovos, D.S., Zschaler, S., Power, C., 2008, "Building Model-Driven Engineering Traceability Classifications", ECMDA Traceability Workshop (ECMDA-TW) 2008, Proceedings, pp. 49–58. Sintef, Trondheim. ISBN 978-82-14-04396-9.

Pramanik,N. 2003, "A Generic Framework For Design Evolution: From Function-To-Form Mapping To Synthesis Of Tolerances", PhD Thesis, Syracuse University, Syracuse, NY.

ProSTEP http://www.prostep.de

Ray, S. R., 2002, "Interoperability Standards in the Semantic Web", Journal of Computing and Information Science in Engineering, March 220, Vol. 2, pp.65-69.

Rezayat, M., 2000, “Knowledge-Based Product Development Using XML and KCs”, Computer-Aided Design, 32, (2000) pp: 299–309

Roy, U. N. Pramanik R. Sudarsan, R. D. Sriram and K. W. Lyons, 2001(a), “Function-to-Form Mapping: Model, Representation and Applications in Design Synthesis”, Journal of Computer-Aided Design, vol. 33, 2001, pp. 699-719.

Roy,U., Pramanik,N. H. Wang, R. Sudarsan, R. D. Sriram and K. W. Lyons, 2001(b), “Tolerance Synthesis Scheme”, NISTIR 6836, Gaithersburg, MD 20899, 2001.

Shooter, S. B., Keirouz, W. T., S Szykman, S and Fenves, 2001, “A Model for the Flow of Design Information in the Open Assembly Design Environment (OpenADE), NISTIR6746, Gaithersburg, MD 20899.

Shooter, S. B., Keirouz, W. T., Szykman, S., and Fenves, S. J., 2000, “A Model for Information Flow In Design”, Proceedings of DETC2000: ASME International Design Engineering Technical Conferences, September 10-13, 2000, Baltimore, Maryland.

Stone RB, Wood KL, 2000, "Development Of A Functional Basis For Design", J Mech Des 122:359-370

Sudarsan R., Mehmet Baysal, Utpal Roy, Sebti Foufou, Conrad Bock, Steven J. Fenves, Eswaran Subrahmanian and Kevin Lyons, 2005, “Information Models For Product Representation: Core And Assembly Models”, International Journal of Product Development, Vol. 2, No. 3, 2005.

Sudarsan, R. Y.H. Han, S.C. Feng, U. Roy, F. Wang, R. D. Sriram and K Lyons, 2003, "Object Oriented Representation of Electro-Mechanical Assemblies Using UML", NISTIR 7057, October 2003, Gaithersburg, MD 20899.

SysML. 2005, "Systems Modeling Language (SysML) Specification Addendum to SysML v. 0.9 Profiles and Model Libraries Chapter", OMG document ad/05-05-01. (http://www.omg.org/cgi-bin/doc?ad/05-05-01).

Szykman, S.; Shooter, S.; Fenves, S. J.; Keirouz, W., 2001, "A Foundation for Interoperability in Next-Generation Product Development Systems" Journal of Computer-Aided Design, 33 (2001) 545-559.

Szykman, S. , Racz, J. W. and Sriram, R. D., 1999, "The Representation of Function In Computer-Based Design" Proceedings of the 1999 ASME Design Engineering Technical Conferences September 12-15, 1999, Las Vegas, Nevada.

Takeda, H.; Yoshioka, M.; Tomiyama, T.; Shimomura, Y.; "Analysis of Design Processes by Functional Evaluation Process"; Analysing Design Activity, pp. 187–209, Chichester (1996), John Wiley & Sons.

Ullman DG., 1993, "The evolution of function and behavior during mechanical design." In: Proceedings of 1993 ASME Design Theory and Methodology Conference, vol. DE-vol. 53. ASME, 1993, p. 91–103.

Umeda, Y., Ishii, M., Yoshioka, M., Shimomura,ıY., & Tomiyama, T., 1996, "Supporting Conceptual Design Based On The Function-Behavior-State Modeler", AI EDAM 10:44, 275-288

Wang, Y. and Nnaji, B.O., “UL-PML: Constraint-Enabled Distributed Design Data Model”, International Journal of Production Research, Vol.42, No.17 (September 2004), pp.3743-3763

Whitney D. E, 2004, “Mechanical Assemblies: Their Design, Manufacture, and Role in Product Development”, New York, Oxford University Press.

Whitney D. E., Mantripragada R., Adams J. D. and Rhee S. J., 1999, “Toward a Theory for Design of Kinematically Constrained Mechanical Assemblies”, the International Journal of Robotics Research / December 1999 pp. 1235-1248

World Wide Web Consortium W3C, http://www.w3.org,

Zha, X.F., Sriram, R.D. and Gupta, S.K., 2005, “Information and Knowledge Modeling for Computer Supported Micro Electro-Mechanical Systems Design and Development”, Proceedings of the ASME Design Engineering Technical Conference, Long Beach, CA, September 24-28,2005.

Vita

NAME OF AUTHOR: Mehmet Murat Baysal

PLACE OF BIRTH: Bremen, Germany

DATE OF BIRTH: February 3rd, 1974

GRADUATE AND UNDERGRADUATE SCHOOLS ATTENDED:

Mechanical Engineering Department, Cukurova University, Adana, Turkey

Institute of Applied and Natural Sciences, Cukurova University, Adana, Turkey

DEGREES AWARDED:

Master of Science (Mechanical Engineering), 2000, Cukurova University

Bachelor of Science (Mechanical Engineering), 1997, Cukurova University

AWARDS AND HONORS:

Graduate Assistantships, Syracuse University 2003-11

Summer Stipend Awards, Syracuse University 2007 - 2010

PhD Scholarship, Scientific and Tech. Research Council of Turkey 2000 - 2002

PROFESSIONAL EXPERIENCE:

Graduate Assistant, Syracuse University, 2009-2011

Mechanical Design Engineer, Emerald Equipment Systems, Liverpool, NY, 2008

Teaching Assistant, Syracuse University, 2004-2005 and 2006-2009

Research Assistant, Cukurova University, Adana, Turkey, 2002-2006

Lecturer, Cukurova University, 1998-2002

Research Assistant, Cukurova University, 1997-1998