CAD/CAPP Integration using Feature Ontology

Christel Dartigues *, Parisa Ghodous **, Michael Gruninger ***, Denis Pallez**, Ram Sriram***

*I3S UNSA-CNRS - 2000, route des lucioles, Les Algorithmes - bât. Euclide B, BP.121, 06903 Sophia Antipolis – Cedex – France
dartigue@unice.fr

**LIRIS, University Claude Bernard of Lyon, 43, Boulevard du 11 novembre 1918
69622 Villeurbanne Cedex, France
{ghodous, dpallez }@liris.cnrs.fr

***NIST, 100 Bureau Drive, Gaithersburg, MD 20899-3460, USA
sriram@nist.gov, gruninger@mie.utoronto.ca

Abstract:
In a collaborative computer-supported engineering environment, the interoperation of various applications will need a representation that goes beyond the current geometry-based representation, which is inadequate for capturing semantic information. The primary purpose of this paper is to discuss a semantically-based information exchange protocol that will facilitate seamless interoperability among current and next generation computer-aided design systems (CAD) and between CAD and other systems that use product data. We describe an ontological approach to integrating computer-aided design (CAD) and computer-aided process planning (CAPP). Two commercial software applications are used to demonstrate our approach. The approach involves the development of a shared ontology and domain specific ontologies in the KIF (Knowledge Interchange Format) language. Domain specific ontologies--which are feature-based—are developed after a detailed analysis of the CAD and the CAPP software. Mapping between the domain ontologies and the shared ontology is achieved by several mapping rules. The approach is validated by using a variety of parts.

Keywords: CAD; CAPP; Interoperability; Ontologies; Design; Process Planning; Knowledge Interchange Format

Introduction

The early part of this millennium has witnessed the emergence of an Internet-based engineering marketplace, where engineers, designers, and manufacturers from small and large companies are collaborating through the Internet to participate in various product development and marketing activities [1, 2, 3, 4]. This will be further enhanced by the next generation manufacturing environment, which will consist of a network of engineering applications, where state of the art multi-media tools and techniques will enhance closer collaboration between geographically distributed applications, virtual reality tools will allow visualization and simulation in a synthetic environment, and information exchange standards will facilitate seamless interoperation of heterogeneous applications. The interoperation of various applications will need a representation that goes beyond the current geometry-based representation, which is inadequate for capturing semantic information. The primary purpose of this paper is to discuss a semantically-based information exchange protocol that will facilitate seamless interoperability among current and next generation computer-aided design systems (CAD) and between CAD and other systems that use product data. Our focus will be on design/process planning integration during the later design stages. We will then present in this paper an approach using a neutral format based on feature ontology. Our work is then decomposed in three main phases that will be further explain in the rest of this paper, as shown in Figure 1:

- The analysis of the two domains studied: detailed design and process planning,
- The creation of the ontology and
- The definition and implementation of mapping rules.

In the next section we provide a brief overview of design/process planning integration. This will be followed by a discussion of representative standards for interoperating design and process planning. The need for ontological approaches is presented followed by descriptions of ontologies in the design and process planning domains and a common ontology. Rules for mapping from and to the common ontology are described. Finally, we illustrate our approach with an example.

FIGURE 1

Design/Process Planning Integration: An Overview

Engineering a product involves several stages with considerable iterations [5]. In this paper we focus on an important aspect of the above cycle: design and process-planning integration. We believe that it is important to integrate design and process planning at various levels of abstraction, as errors made during early design stages could have a significant impact on the overall product quality and costs [5-8].

Engineering design involves mapping a specified function (or functional specifications) onto a (description of a) realizable physical structure – the design artifact. Over the past several decades considerable research has been done in developing various design product and process models [9]. We will not delve into a detailed description of the design process, much as we feel a need for the adequate representations for process knowledge. The reader is referred to [5] for a formal description of a design process model. At this stage our primary concern is on the product or artifact representation. For this we use the NIST CORE product model presented in [10].

Process planning is an intermediate phase between design and manufacture [11, 12]. More precisely, it links these two decisive phases of product development [13]. It depends on choices made in design and determines precisely actions that will be achieved during manufacture. Different definitions have been given for process planning [7, 14, 15, 16]. We use the following one in this paper: process planning is the phase that, from information generated during preliminary design (product geometry for instance), determines necessary operations and actions to transform a raw part in a finished or semi-finished part, the necessary human and material resources to manufacture the product, as well as the product development cost evaluation.

A wide variety of manufacturing processes are available for the actual artifact production. In the current work we focus on the machining processes for part production, in particular material cutting processes. Figure 2 provides a representation of this process: the cutting tool comes against the surface, creating a chip that will be removed from the part.

FIGURE 2

The interactions between design and process planning occurs at various stages, from conceptual to detailed design/process planning as shown in Figure 3 [17]. Current interfaces between design and process planning are defined during the detailed design stage. This is primarily achieved through use of geometric features. However, there is considerable difference in the methods and terminology used: features are used to design a product (design by feature) [8, 13, 18, 19] while in process planning they are extracted from the product (feature recognition or extraction) [13, 18, 19, 20, 21], and a consistent feature terminology does not exist for the two domains. These different viewpoints of designers and process planners on features makes data exchange a tedious task. Although features are considered differently in design and process planning, they represent a natural link between these two domains. Hence, features provide a valuable mechanism for information exchange. Next we review the current standards in design and process planning interoperability and discuss extensions needed for feature-based interoperability.

FIGURE 3

Standards for Interoperability

We illustrate the interoperability issue between CAD systems by considering a potential information exchange scenario during the design of the Boeing 777. For Boeing to incorporate Rolls Royce engines into the design, the data format has to be converted from Computer Vision's CADDS (used by Rolls Royce) to Dassault's CATIA. Similarly, for Rolls Royce to understand changes made by Boeing engineers, the data need to be converted from CATIA to CADDS. Hence, we need at least 2 translators. For three systems this grows to 6 translators and for n systems we need n(n-1) translators. Hence, there is a need to design, build, and maintain n(n-1) translators. A solution to this problem is to use a neutral format and make all the CAD applications output this format. Doing so will reduce the number of translators to 2*n, i.e., for each CAD system we will need two translators -- one from the CAD system to the neutral format and the other from the neutral format to the CAD.

A standard of primary interest to design is ISO 10303, also known informally as STEP [3, 34, 23] (Standard for the Exchange of Product model data) and developed by the International Organization for Standardization Technical Committee 184/ Subcommittee SC4 (ISO TC 184/SC4). Its intention is to enable the exchange of product model data between different modules of a product realization system, or the sharing of that data by different modules through the use of a common database [24]. The first parts of STEP to achieve International Standard status were published in 1994, but many other parts have since been published or are under development and will eventually be added to the standard. Recent updates (and other relevant details) can be found at the following websites: http://www.nist.gov/sc4, www.tc-184-sc4.org, and http://www.iso.ch/iso/en/ISOOnline.frontpage.

Considerable research has been performed on mapping CAD data onto process planning systems. However, this work has met with limited success, such as the one reported by [25]. One problem with the current standards is the lack of integration between CAD data output and process planning input. For example, the primary focus of STEP AP 203 is the interoperability between geometry-centered CAD systems, while the focus of STEP AP 224 (Mechanical product definition for process plans using machining features) has been on input to process planning systems with a primary focus on representation of machine features. The idea of features has been in vogue for some time and the literature is abound with definitions of features [15, 19, 26-32]. For example, Shah et al. suggest that features "are primitive or low level designs with their attributes, qualifiers and restrictions which affect functionality and/or manufacturability. Features can describe form (size and shape), precision (tolerances and finishing), or materials (type, grade, properties and treatment), and vary with product and manufacturing process".

To achieve truly collaborative design and engineering, exchange representations of both design and process information must support multiple levels of abstraction. To adequately achieve this we will need a more formal method for representing features, such as the ontological approach described in the next section. Our approach has some similarities to the one presented in [33], but our overall methodology is different.

Ontological Approach to Interoperability

In all types of communication, the ability to share information is often hindered because the meaning of information can be drastically affected by the context in which it is viewed and interpreted. This is especially true in manufacturing, because of the growing complexity of manufacturing information and the increasing need to exchange this information among various software applications. Different representations of the same information may be based on different assumptions about the world, and use differing concepts and terminology -- and conversely, the same terms may be used in different contexts to mean different things. Often, the loosely defined natural-language definitions associated with the terms will be too ambiguous to make the differences evident, or will not provide enough information to resolve the differences.

To address these challenges, various groups within industry, academia, and government have been developing sharable and reusable models known as ontologies [3]. All ontologies consist of a vocabulary along with some specification of the meaning or semantics of the terminology within the vocabulary. In doing so, ontologies support interoperability by providing a common vocabulary with a shared semantics. Rather than develop point-to-point translators for every pair of applications, one simply needs to write one translator between the application's terminology and the common ontology. Similarly, ontologies support reusability by providing a shared understanding of generic concepts that span across multiple projects, tasks and environments.

The various ontologies that have been developed can be distinguished by their degree of formality in the specification of meaning. With informal ontologies, the definitions are expressed loosely in natural language. Semi-formal ontologies, such as taxonomies, provide weak constraints for the interpretation of the terminology. Formal ontologies use languages based on mathematical logic. Informal and semi-formal ontologies can serve as a framework for shared understanding among people, but they are often insufficient to support interoperability, since any ambiguity can lead to inconsistent interpretations and hence hinder integration.

Another source of semantic heterogeneity lies in the languages used to represent the ontologies. There have been several efforts within academia and industry to develop common languages that can be used as the basis for ontologies to support semantic integration; the most expressive is the Common Logic project, which combines the Knowledge Interchange Format [34-36] and Conceptual Graphs (CG) [37] languages. Common Logic includes a core language that has the expressiveness of first-order logic; its syntax and semantics are those of traditional first-order logic. Some other languages has been based on Logic, such as PSL [38]. Most recently, this has been extended to include extensions that allow sorted formulae for the specification of class hierarchies, and the specification of the meta theory of KIF within the language itself.

Our objective in this paper consists in developing and implementing an approach for data exchange between designers and process planners. To realize this, we decided to develop a feature ontology. This ontology will represent all the common knowledge between designers and process planners as well as specific knowledge of both experts. We will use this ontology as depicted as follow: a designer creates an artifact shape model using CAD software (such as Pro/Engineer); this model is then transformed, using mapping rules (see Section 8), into instances of the shared ontology. These instances of the shared ontology are then transformed, using other mapping rules, into a representation interpretable by CAPP software (such as Pro/Engineer). Features represent for us a common knowledge that will be the base of our shared ontology for date exchange. In the next parts of this paper, we will present the design specific parts, the process planning specific parts and the design and process planning common parts of our ontology. We will continue with the description of the mapping rules used to translate data and we will finished by an example.

Design Feature Ontology

Our ultimate goal is to develop a comprehensive feature model that can be used through the entire design life cycle. However, for our prototype we restricted the NIST CPM’s extensions to the information generated by commercial CAD systems. To identify these concepts, we first performed an extensive analysis to understand various designers' needs. This analysis phase involved:

- The extraction of designer know-how--which is implicit--in order to formalize designer's knowledge; and
- The analysis of different CAD software such as Pro-Engineer and SolidWorks: we used them to create various parts in order to better understand the design process.

Based on this analysis we concluded that the NIST CPM had most of the necessary classes to represent detailed design data. We added a few classes in order to increase the coverage to CAD software, such as: the datum coordinate system in which the artifact is defined, the dimensions associated to an artifact, the precision of the dimensions of an artifact, the different versions of an artifact and the constraints associated to each feature. Figure 4 represents these concepts.

FIGURE 4

We also defined different kind of constraints as shown in Figure 5. The initial categories that we considered are position and orientation constraints, which can be further classified into attachment and geometric constraints. Attachment constraints specify how a feature instance is attached to the global model by coupling some of the feature faces with the pre-existing faces. Geometric constraints specify geometric relations such as parallelism of two faces or distance between two faces. Validity constraints correspond to another constraint category defined in our ontology. These validity constraints can be further classified into dimension constraints, algebraic constraints, boundary constraints and feature interaction constraints.

FIGURE 5

The above extensions suffice to illustrate our approach. Additional classes will be needed for a wider coverage. KIF representations of a representative set are shown in Figure 6.

FIGURE 6

Process Planning Ontology

Our feature ontology is also representative of the process planning viewpoint. We followed a similar approach used for design: we asked process planners to describe how they work, what kind of information they need, what are the different phases of their work, etc. We also studied CAPP software: PART. This analysis of process planning turned out to be a more difficult task than obtaining the design features. While designers have a consistent notion of what design is, process planners seem to be in less agreement on the terminology in their domain. Based on our discussions, we decided to use the concepts presented in Figure 7.

FIGURE 7

In this figure, an artifact is associated with a manufacturing model. This model is used to create a process plan.

The input of this process plan is a raw part and the output is a semi-finished or finished part. A process plan identifies the machining operations that are necessary to manufacture an artifact. Hence, a process plan is composed of machining setups, which contains all the machining operations that are realized with the same machine and without changing the attachments. For each machining setup, there is a set of machining operations. Each machining operation is then realized with the same machine and attachments. Each machining operation is composed of a set of machining sequences, which corresponds to a transformation of a part that is achieved with the help of a material removal tool moving according to a tool path. Finally, a machining operation modifies a surface in accordance to a required finish: raw, semi-finish, finish or super-finish.

Common Feature Ontology

The last part of our ontology corresponds to the common concepts between design and process planning and is composed of numerous classes and relationships. We base our ontology on the NIST Core Product Model (CPM). We used this model in order to take into account general concepts, initially present in this model, and we added more specific concepts allowing feature representation. Figure 8 represents the main classes and relationships composing the Core Product Model and its extensions in this work, where the extensions are shown as darkened boxes (ideally, the NIST CPM should be a package in UML and our extensions should be in a separate package). The descriptions of key entities in the NIST CPM are as follows (taken from [10]).

FIGURE 8

An Artifact represents a distinct entity in a design, whether that entity is a component, product, subassembly or assembly. The Artifact's attributes refer to the Specification responsible for the Artifact and the Form, Function, and Behavior comprising the Artifact. The Function represents what the Artifact is supposed to do. The Artifact satisfies the engineering requirements largely through its Functions. The Form of the Artifact can be viewed as the proposed design solution for the design problem specified by the Functions. More precisely, the physical characteristics of an Artifact are represented in terms of its Geometry and Material properties.

Another important class of the Core Product Model is the Feature. An Artifact is composed of a set of features, where a feature is a subset of the form of an object that has some function assigned to it. We can have several types of features: analysis features, design features, manufacturing features, interface or port features, etc., Compound features can be generated from primitive features. The notion of a feature is further elaborated in the work presented here.

FIGURE 9

We modified the NIST Core Product Model (CPM) by adding some concepts that are common to design and process planning, are both necessary for designers and process planners, and are considered in CAD and CAPP software.

Our main objective is to find a common feature representation between design and process planning. To do so, we extended NIST CPM to address the following:

- The way each feature is represented, such as a B-Rep representation, a CSG representation, a swept representation, etc. (Feature Representation concept); and
- The elements composing each feature, such as a bottom side, an intermediary face, etc. (Feature Element concept).

We also characterized a complete feature decomposition which is based on the feature categories proposed in the part 48 of STEP [40]. Figure 9 illustrates this decomposition. Features are classified into: volume features, transition features, and pattern features. A more detailed description of this decomposition can be seen at [40, 41].

Mapping Rules For Case Study

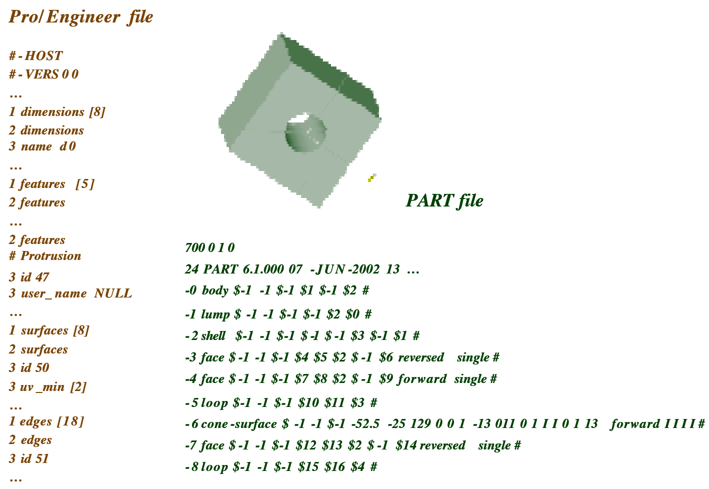
Once the feature ontology in various domains is defined, the next step is to define the mapping rules that will transform specific files onto instances of our common ontology. For our case study, we choose the following software: Pro-Engineer, which is used by CAD experts, and PART, which is used by CAPP experts. The methodology that we followed is described in Figure 10.

FIGURE 10

We first analyzed the existing export and import formats of Pro-Engineer and PART. Then, we selected one format for each of them: a proprietary format for Pro-Engineer, called Neutral File Format, and ACIS format for PART. Once the formats have been chosen, we analyzed the representation of different artifacts in the two formats to extract all the important concepts represented in each file in order to correlate them with the domain ontology entities. We then defined two algorithms: one to translate a file generated by CAD software into a set of instances of the feature ontology and one to translate this generated file into a file that can be interpreted and processed by a CAPP software. The inputs to the first algorithm are the file containing the entire description of the ontology, which is expressed in KIF, and the file generated by the CAD software (Pro/Engineer in this case), which represents the geometry and topology of the part that has to be manufactured. The inputs to the second algorithm are the file containing the entire description of the ontology, which is the common ontology expressed in KIF, and the file generated by the first algorithm.

As we previously stated, the only assumption made during the elaboration of the ontology and the mapping rules was that we only considered parts that do not have any assembly; solving this problem for simple machining parts containing only features by itself is a difficult task. Taking into account more complex parts containing for example assemblies would implies to modify both the ontology and the mapping rules.

FIGURE 11



Pro/Engineer file

```
#-HOST
#-VERS 0 0
...
1 dimensions [8]
2 dimensions
3 name  d0
...
1 features  [5]
2 features
...
2 features
# Protrusion
3 id 47
3 user_name  NULL
...
1 surfaces [8]
2 surfaces
3 id 50
3 uv_min  [2]
...
1 edges [18]
2 edges
3 id 51
...
```

PART file

```
700 0 1 0
24 PART 6.1.000 07  -JUN-2002  13  ...
-0  body $-1  -1 $-1 $1 $-1 $2 #
-1 lump $ -1 -1 $-1 $-1 $2 $0 #
-2 shell  $-1 -1 $-1 $-1 $-1 $3 $-1 $1 #
-3 face $-1 -1 $-1 $4 $5 $2 $-1 $6 reversed   single #
-4 face $-1 -1 $-1 $7 $8 $2 $-1 $9 forward  single #
-5 loop $-1 -1 $-1 $10 $11 $3 #
-6 cone-surface $ -1 -1 $-1 -52.5  -25 129 0 0 1  -13 011 0 1 I I 0 1 13   forward I I I I #
-7 face $-1 -1 $-1 $12 $13 $2 $-1 $14 reversed   single #
-8 loop $-1 -1 $-1 $15 $16 $4 #
```

Figure : Data declaration in Pro-Engineer and PART files for a simple artifact

For a simple artifact such as a box with one hole (Figure 11), the file generated by Pro-Engineer is hierarchically structured: it contains the dimensions characterizing the artifact, the features used to build it, the surfaces determining the features and the edges composing the surfaces. PART files are totally different: information is stored with no specific order, and data contained in such files relates to geometric and topologic information. This kind of file format doesn’t explicitly provide information about features composing an artifact.

Using different instances of Pro-Engineer and PART files, we extracted a list of entities or concepts and their attributes in these files, such as: plane surface, cylindrical surface, straight curve, linear curve, edge, point, vertex, etc. Once this analysis is done, we elaborated the mapping rules between a Pro-Engineer file and a file containing instances of our ontology and between this generated file and a PART file. The objective of these rules is to identify in our domain ontology the entities that are equivalent to the concepts identified in Pro-Engineer and PART files. We first expressed these mapping rules graphically. Figure 12 shows the graphical representation of one such mapping rule, which shows the correspondence between a plane surface expressed in a neutral file generated by Pro-Engineer and the equivalent concepts in our ontology.

FIGURE 12

Once this step is finished, we obtained two sets of mapping rules. These mapping rules are expressed graphically. The next step consists in implementing these rules in order to be able to translate a CAD file into a CAPP file via our ontology. As we have previously stated, our method involves starting from a file generated by Pro/Engineer, applying a first set of mapping rules in order to generate a neutral file, and applying our second set of mapping rules on this neutral file in order to obtain a file interpretable by PART. A description of our mapping algorithm is shown in Figure 13.

FIGURE 13

Starting from a CAD file generated by Pro/Engineer, we create all the features. For each feature, we have to extract from our ontology all the attributes that we have identified for a feature (for example the list of surfaces, the list of dimensions, etc.). For each of these attributes we search, still in our ontology, the nature of the attribute, which can be either simple (i.e., integer, string, boolean) or complex (i.e., the attribute is composed of sub-attributes). If the attribute is a simple one, we extract in the initial file the associated value and we add a new instance in the neutral file. If the attribute is more complex, we consider each sub-attribute until all concepts appearing in the initial file have been instantiated. The advantage of this algorithm is that if we decide to change the attributes of one of the concepts of the ontology –for example if we delete one attribute of the concept feature- the algorithm will not have to be change because the number of attributes of a concept is calculated each time the algorithm is running.

We tested our methodology and our prototype with different examples. For our initial prototype we considered only simple parts (see Figure 14), with great success. Our plans are to extend this work for complex artifacts (e.g., assemblies).

FIGURE 14

Summary

In this paper we have described an ontological approach to integrating computer-aided design (CAD) and computer-aided process planning (CAPP). Two commercial software applications were used to demonstrate our approach. The approach involved the development of a shared ontology and domain specific ontologies in the KIF (Knowledge Interchange Format) language. Domain specific ontologies--which were feature-based--were developed after a detailed analysis of the CAD and the CAPP software. Mapping between the domain ontologies and the shared ontology was achieved by several mapping rules. The approach was validated by using a variety of parts.

Disclaimer

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

Bibliography