

Features and Interoperability of Computer Aided Engineering Systems

Yanan Xie, Jikai Liu, Hongyi Liu and Y.-S. Ma

1 CAx Systems, Customization, and Application Development

1.1 Introduction to CAx Systems

CAx is a broad term that means the use of computer technology to aid in the design, analysis, and manufacture of products. CAx usually includes computer-aided design (CAD), computer-aided engineering (CAE), computer-aided manufacture (CAM), computer-aided process planning (CAPP), and product data management (PDM) [78].

In the design process, increasingly more tasks have been supported by CAx tools in the last 30 years. Starting with drafting and surfacing, classical mechanical design was gradually replaced by 3D wire frame, solid modeling, and parametric and feature-based design. Today the entire product creation process, including production preparation, is run with CAx. According to the various application stages, CAx systems were developed with different computer solutions, such as computer-aided styling (CAS) [98], computer-aided esthetic design (CAAD) [76], computer-aided conceptual design (CACD) [97], and so on. All these technologies are categorized as different aspects of CAD/CAM and CAE, two of the more important CAx technologies, were developed almost independently. The latter is mainly used in a limited sense for simulation and finite element analysis (FEA). Although started independently as separate packages, both technologies require geometry data input from CAD.

Y. Xie · J. Liu · H. Liu · Y.-S. Ma (✉)
Department of Mechanical Engineering, University of Alberta,
Edmonton, AB T6G 2G8, Canada
e-mail: yongsheng.ma@ualberta.ca

1.2 Function and Data Management of CAx

Advanced CAx tools merge many different aspects of product lifecycle management (PLM), including design, FEA, manufacturing, production planning, virtual product testing, product documentation, and product support. With the growing integration of these CAx tools, data and information management has become increasingly important to realizing those expected industrial benefits. Currently, the complex network of CAx systems and their various data cannot be handled without a product data management system (PDMS). PDMS was regarded as the backbone of modern product development and now is extended to support the whole product lifecycle. This new paradigm of coherent multistage and multi-view information management has led to a wave of research effort labeled PLM.

1.3 Main CAx Software Tools

CAx software tools have been produced since the 1970s for a variety of computer platforms. A landscape of the main CAx software tools is shown in Table 1. A kernel is the brain of the CAD application. A modeling kernel is a collection of classes and components that comprise mathematical functions performing specific modeling tasks [102]. Currently, in industry, CAD applications are usually generated from a commercially available kernel. AutoCAD, NX [64] and CATIA [13] use their own kernels, while most other applications use either ACIS from Dassault Systèmes [86] or Parasolid from Simence PLM [64].

Table 1 The main CAx software tools

| | |
|-----|-------------------|
| CAD | AutoCAD |
| | Autodesk inventor |
| | NXCAD |
| | Solid edge |
| | Pro/ENGINEER |
| | CATIA |
| | Solid works |
| | NX Nastran |
| | CATIAABAQUS |
| | NXCAM |
| CAE | CATIA |
| | Team center |
| CAM | Windchill |
| | |
| PDM | |
| | |

1.4 Customization

As CAx systems are so widely used in nearly every industry, deploying the right computer solution for each aspect of the engineering workflow demands exact data structure matching for information exchange and well-planned procedures to streamline the execution of computer functions. Industries that produce medical devices, machine tools, apparel, as well as those specialized engineering fields such as metrology and ship-building, are characterized by the need for special CAx software for specific functions. Rather than using the common “as-is” versions of CAx software tools, progressive companies often develop their own versions as a way to implement the required differentiation in the product development cycle. Such customized solutions can accelerate new process chains, and improve the final customer experience. Many software platforms like NX (see Fig. 1) and CATIA [13] offer customized solutions for specific CAx process chains.

The hierarchy within the CAx system is shown in Table 2, which illustrates the four levels of composition in the typical CAx application chain. Levels 1 and 2 are developed by various vendors as packaged commercial products. They are vendor-dependent and do not differ much according to customer requirements, with only limited customization features for user interfaces and user-defined templates. Advanced solutions are categorized as “extended application modules” in level 3 and “tailored solutions” in level 4.

In level 3, users select those extended application modules offered by the vendors according to more specific application areas. For example, MoldWizard will be selected for plastic injection mold design. There are hundreds of choices in this level: within the Siemens NX suite alone, there are many such modules offered, including machining application modules such as 3-Axis Machining, 5-Axis Machining, CAD for Numerical control (NC) Programming, Data Exchange, High Speed Machining, Machining Simulation, Multi-Function Machining, NCDATA Management, Part Planning, Post Processing and Post Processor Library, Programming Automation, Resource Management, Shop Documentation, Wire EDM, and more [64].

Most platforms offer open API to support secondary development. The customization in level 4 therefore mainly comes from the connection with customized solutions. The development of customized solutions depends on the real needs of consumers and usually has an evident economic advantage. Further discussion on this topic can be found in Sect. 1.5.

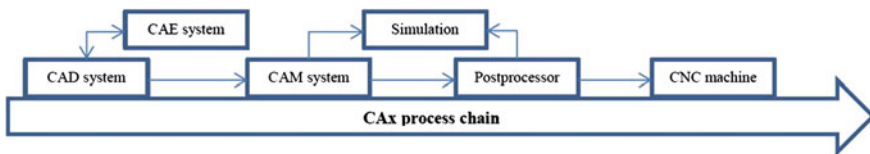


Fig. 1 Partial NX CAx process chain

Table 2 Hierarchies of a CAx system

| Level | Composition | Description |
|-------|------------------------------|--------------------------------------|
| 4 | Tailored solutions | Specific to end-user companies |
| 3 | Extended application modules | CAM for milling, mold wizard, etc. |
| 2 | CAD/CAM/CAE | Platforms such as NX, Pro-E |
| 1 | Kernel | Core kernels like parasolid and ACIS |

1.5 Application Development

Application development based on the CAx system is a programming- and research-intensive process. Numerous applications have been developed and widely used in recent years, but this still cannot satisfy users' needs. Currently, most CAx software packages offer application programming interfaces (APIs) to satisfy application development needs [66]. An API is a source code-based specification intended to be used as an interface by software components to communicate with each other. An API may include specifications for routines, data structures, object classes, and variables. For example, CATIA and NX both offer their own open API for application development.

CATIA V6 can be adapted using Visual Basic and C++ programming languages via component application architecture (CAA). CAA is Dassault Systèmes's comprehensive, open-development platform that enables developers to integrate their solutions. This collaboration expands Dassault Systèmes's system offerings and gives customers a larger set of CAx solutions to meet their specific industrial needs [14].

Pro/TOOLKIT is an API that allows Pro/ENGINEER functionality to be augmented and/or customized to meet the specific needs of PTC's customer base by using the "C" programming language. Specifically, Pro/TOOLKIT provides the ability to customize the standard Pro/ENGINEER user interface, automate processes involving repetitive steps, integrate proprietary or other external applications with Pro/ENGINEER, and develop customized end-user applications for model creation, design rule verification, and drawing automation [71].

2 Interoperability Among Systems

As part of the trend toward customizable and flexible product development tool suites, multiple software tools—even from different vendors—are often used for various phases of product development. As a result, the problem of interoperability among systems emerges. This reflects the unfortunate reality that software vendors tend to use proprietary data representation as a competitive advantage, which severely inhibits interoperability.

To solve the interoperability problem, it is necessary to identify two macro domains of application: horizontal data exchange and vertical data exchange [7]. Horizontal data exchange is taken to mean data exchange between different CAD systems, mainly focusing on geometric information. However, more important than just geometric information is the design intent, which is stored in design history and constraints. This makes the design intent reservation during data exchange something of a hot issue. At present, some commercial tools for CAD geometry data exchange are available, such as a conversion engine in CrossCAD [96], which can import, analyze, heal, and export models across CAD systems.

Currently, it is common practice to have a design geometry data model created from CAD systems translated into an intermediate data format like the STEP file format, and then imported into CAE and CAM applications. Conceptually, such data exchange is referred to as vertical because the data is transferred from the upstream application into downstream applications. In selection of the intermediate data format, there are two options: a proprietary data format or a neutral data format (NDF) [99]. Most commercial data exchange service providers tend to use a proprietary data format, which offers a competitive edge. For instance, NX and its PLM) solution Teamcenter are aimed at offering a complete solution from design to manufacture [64]. In contrast, an increasing number of industry companies have adopted neutral formats for data exchange; in the course of data exchange technology development history, several international standard data formats were proposed, such as IGES, PDES, and STEP [67]. Among these standards, STEP is the most advanced and complex, covering almost all the applications used in each product lifecycle phase. This topic is discussed in more detail in Sect. 3.

In contrast to file-based data exchange, recent research [7] has attempted to create a flat interapplication data service scheme that enables various engineering applications to share their models via the use of API functions. This approach is referred to as interface-based horizontal data exchange (see Fig. 2). Typically, client-server architecture is used for such a system: the CAD system provides its functions and data models via a coordination server, while the downstream applications receive services as subscribers. Bianconi et al. [7] summarize the advantages of such a system as data centralization, synchronization, and encapsulation.

2.1 Review of Interoperability and Related Technologies

The interoperability gaps among different computer-aided tools are well recognized across engineering domains, which call urgently for systematic integration to enhance interoperability and, hence, benefit the lifecycle. From the point of view of concurrent engineering, interoperability among applications can be enhanced on three levels: knowledge, information, and data [61]. As illustrated in Fig. 3, a NDF (such as IGES, STEP, or IDEF) provides standards to facilitate data sharing and

Fig. 2 Interface-based horizontal data exchange [7]

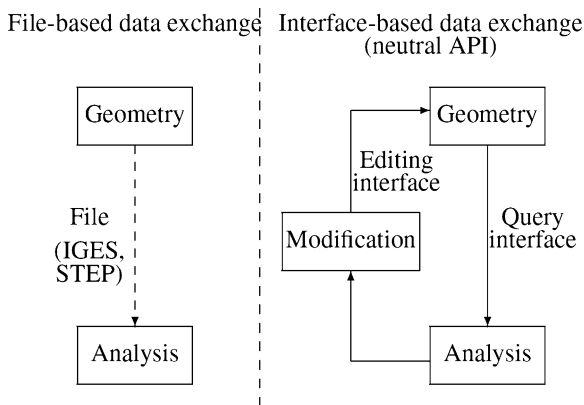
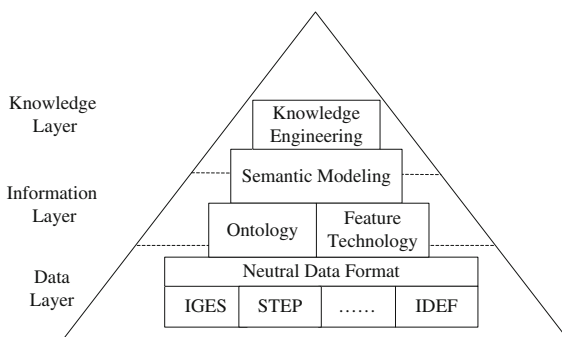


Fig. 3 Data representation pyramid for interoperability



exchange from the bottom data layer. Semantic modeling, a methodology that can effectively support knowledge engineering and feature knowledge, offers a flexible and scalable way to enhance interoperability.

2.2 Neutral Data Format

From the data layer, NDF provides a standardized intermediate data model to facilitate data exchange and sharing. An illustration of data transfer via NDF between computer applications among various domains is shown in Fig. 4. The purpose of NDF is to transfer data from all applications into a NDF, which requires the development of pre- and post-translators for each computer system involved to enable data transfer [67]. This significantly reduces the number of interfaces needed, as well as development effort and maintenance complexity. Based on this neutral data translation approach, any future potential development of more advanced application integration within a broader collaboration environment will be made feasible and efficient, as only the interface between NDF and the new

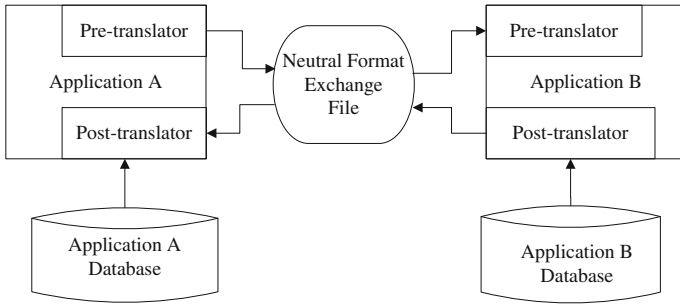


Fig. 4 Data transfer between computer-aided tools

application needs to be developed. Three foremost NDFs are IGES, STEP, and electronic design interchange format (EDIF), which are elaborated in the following subsections.

2.2.1 Initial Graphics Exchange Specification

The Initial graphics exchange specification (IGES), the foremost NDF, is a standard for graphics information exchange between CAx systems. It is designed to be independent of any computer systems but is capable of capturing all the information existing in the CAx applications, including binary information, start, global, directory entry, parameter data, and terminate sections [67].

Although efforts spent on improving capability with solid modeling has gained some results in IGES versions 4.0 and 5.0, the deficiency in solid modeling is still not significantly improved, which always leads to loss of information during the process of data exchange and sharing. The existence of the Standard for the exchange of product data model (STEP) reduced the urgency of further development and made IGES version 5.3 the last published standard in its series in 1996.

2.2.2 Standard for the Exchange of Product Data Models

STEP (ISO 10303), a standard for the representation and exchange of engineering product data, makes it possible to develop a complete and integrated product description in a NDF and, hence, to facilitate interoperability among different computer-aided systems throughout the product development lifecycle [40]. The standard is also known as the STEP. It is organized in a series of sections, or application parts (APs), covering the representation of product information (including components and assemblies) as well as the exchange of product data, which provides the capability of describing data throughout the lifecycle independently of any particular computer system.

STEP was developed as an alternative to IGES and boasts a more comprehensive set of definitions [87] for a set of neutral product information entities, especially for geometric ones. STEP provides a mechanism that describes a complete and unambiguous product definition throughout the lifecycle of a product, independent of any computer system. This international standard is accepted by most vendors, so it is quite suitable for use in realizing data interoperability. Users can implement proper APs to meet their product data exchange requirements [39]. Some APs are listed in Table 3, with their roles in integrating manufacturing activities shown in Fig. 5. Users can implement proper APs to meet their product data exchange requirements [39, 40].

APs across the disciplines of chemical, mechanical, and electrical engineering are illustrated in Fig. 6. AP 221, “Functional Data and Their Schematic Representation for Process Plants,” specifies an exchange scheme that is applicable to chemical process projects [46]. The scheme describes the data structures used for communicating functional design and engineering specifications of system components, which can also be used for subsequent procurement and component manufacturing generally carried out by engineering, procurement, and construction (EPC) companies. Reference data, which comprises instanced templates in the form of library elements, is designed to be referenced together with the data structure specifications to facilitate collaborative system design across disciplines [3].

Another important application protocol, AP 227 (“Plant Spatial Configuration”), provides a standard for exchange among engineers from different disciplines as well as operation owners and EPC companies during the lifecycle of chemical process projects. In this AP, the information requirements for the exchange of design and layout models of a process plant are specified. It also specifies other integrated engineering resources, such as those models required for design, analysis, and fabrication of piping components and piping systems [44]. The exchange of functional characteristics of heating, ventilation, and air conditioning (HVAC), mechanical and piping components and systems, and schematic representation are also addressed. Similarly, it is specified in AP 231, “Process Engineering Data: Process Design and Process Specifications of Major Equipment,” that the representation of process steps involved in a chemical process be

Table 3 STEP application protocols

| AP | Title |
|-------|---|
| AP204 | Mechanical design using boundary representation |
| AP207 | Sheet metal die planning and design |
| AP209 | Composite and metallic structural analysis and related design |
| AP219 | Managing dimensional inspection of solid parts or assemblies |
| AP223 | Exchange of design and manufacturing product information for cast parts |
| AP224 | Mechanical product definition for process planning using machining features |
| AP238 | Application-interpreted model for computerized numeric controllers |
| AP240 | Numerical control process plans for machined parts |

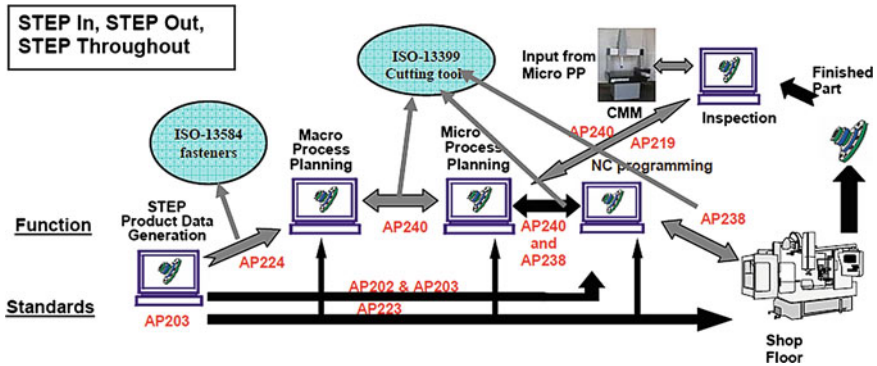
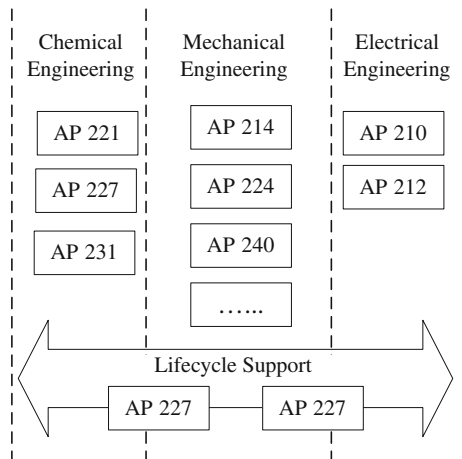


Fig. 5 Data exchange APs based on STEP [39]

Fig. 6 Application protocols providing lifecycle support



included, along with material and reaction data, process flow diagrams (PFDs), and detailed process and plant descriptions [3].

Within the mechanical engineering domain, there are even more APs defined in STEP, such as APs 203, 204, 214, 224, and 240. Application protocols for information representation and data sharing within mechanical engineering among CAx systems are addressed in these APs. This provides a neutral file exchange support independent of any particular computer-aided system.

Thanks to the collaboration between the International Organization for Standardization (ISO) and the International Electrotechnical Commission (IEC), STEP also defines such APs as AP 210 and AP 212, which are applicable to projects that span both mechanical, electrical, and electronic engineering domains. AP 210, “Electronic Assembly, Interconnect, and Packaging Design,” specifies the transformation from detailed requirement data (such as functional descriptions of the device, manufacturing processes required and other design specifications) into data

structures and formats that are analyzable and processable by the manufacturing systems, which will facilitate information-sharing between engineers from different domains [48]. However, this AP is not limited to the electronic domain. Another AP specified in the electronic domain, “Electrotechnical Design and Installation,” (AP 212) specifies the information shared between the systems involved in the design and installation as well as the commissioning of electrical equipment [48]. The data specified in this AP, such as that which pertains to equipment design and installation, will effectively enhance interoperability across domains.

AP 232, “Technical Data Packaging Core Information and Exchange”, addresses the packaging of products for exchange as well as the exchange requirements of product data groupings [42]. AP 239 is another member of the application protocol series specified for product lifecycle support, which can be applied to developing an integrated series of interfaces to provide data interchangeability within one domain or across disciplines [45]. These APs can be used to support the lifecycle of chemical process engineering projects (from process conceptual design, engineering analysis, and mechanical engineering and design) to implementation and maintenance, which involves the disciplines of chemical, mechanical, and electrical engineering.

Although STEP has contributed considerably to interoperability, it still suffers from the complexity of implementation in real applications, which requires too much information modeling and development [61]. Restricted by the APIs of commercial CAD systems, which are not designed for model exchange, information associated with the models, including the design intent, is very likely to be stripped from the data during the exchange process [52]. Hence, the stalled effort toward interoperability needs the introduction of a new technology, such as feature technology.

3 Current Standards’ Limitations

Among the standards established for product information exchange, IGES and STEP, which have been set up and maintained by American National Standards Institute (ANSI) and ISO, respectively, are the most powerful and widely accepted. In this section, the advantages and limitations of both are described in detail.

IGES, a data format depicting product design and manufacturing information, mainly assists data exchange between CAD and CAM systems. IGES is independent of all CAD and CAM systems, and is thus an NDF [67]. As IGES was established in 1980, it has some (if limited) capability for the exchange of points, edges, and surface entities, but cannot handle solids at all; IGES, therefore, cannot support the full range of CAX entity chain processes. Ideally, data exchange should be supported across the product lifecycle [51]. Although IGES has obvious drawbacks, it still has 20 % of the usage level in the CAD data exchange field in

North America due to the simplicity of its implementation. In comparison, STEP only has about 15 % [37].

The best-developed application of STEP is the achievement of CAD/CAM and computer numerical control (CNC) integration. The relevant standard is ISO 6983 [38]. Recently, new standards have been developed, such as ISO 14649 [43] and ISO 10303 AP 238 [47]. They provide CAM and CNC vendors the opportunity to develop highly intelligent CNC controllers, which can realize bidirectional communication of standardized geometric and manufacturing data in the form of features [106].

For future research, new standards or new versions of existing standards should better support engineering semantics. Product ontology representation should be exchangeable for interoperability among information systems across the product lifecycle [90]. We can see that IGES and STEP have a common limitation in their inability to transfer design intent, such as construction history and constraints. Research efforts [52, 70] are increasingly devoted to this issue, but it has still not been adequately addressed.

4 Feature Technology

Feature-based product modeling was traditionally used for geometrical construction with certain predefined templates, and most CAD tools embraced this approach to facilitate interaction with designers. In this kind of CAD product model, geometric features are the basic components for building up the shape: a variety of features ultimately constitute a complete product in a hierarchical structural model. A *feature* encapsulates the engineering significance of a portion of the physical constitution of a part or assembly, and hence is important in product design and definition for a variety of computer-aided systems [77].

Generally, the feature-based approach uses a set of basic features as a starting point, then adds other advanced and user-defined features to enhance application-specific knowledge encapsulation and process automation. Moreover, if features are integrated with parameters and other features, they can be tracked. Numerous research efforts have been devoted to the feature-based design approach [108]. Monedero gives a basic definition of the integration of parametric design and modeling [63].

Ideally, when a feature's parameters change in a feature-based system, the other related features will be changed accordingly. This is the functional superiority of such an associative feature approach as compared to other procedural approaches. With the progress of feature technology, research has merged into two mainstream methodologies: Design by feature (DBF) and Feature recognition (FR) [11, 79].

DBF is a design modeling method used for pattern-based functionality and manufacturing geometric entities, in which the model is built in terms of features provided by an existing feature library [1]. One of the major challenges in applying the DBF method is that the limited and rigid definitions of available features

constrain the creativity of designers. It is also impossible to predefine all design and manufacturing features.

FR is a method from an opposite perspective: instead of designing from features, it aims to recognize features from an existing geometry model. This method is mainly used for manufacturing purposes after CAD design models have been created and before CAM tool path generation is applied based on said models. FR is further elaborated in [Sect. 5.1](#).

5 CAD/CAM Integration via Features

As yet there is still no consensus on a definition of a feature, but feature technology has been widely applied in the integration between CAx systems, such as CAD/CAM and CAD/CAPP [2, 109]. Feature-based CAD/CAM integration is a technology used to realize automatic transmission and conversion of product information among CAD, CAPP, and CAM systems [29, 78]. While it is true that the CAD/CAPP/CAM systems are at a maturation stage individually in their traditional functions; but because they have been developed separately, they emphasize their separate functionalities. They use different data models and formats, which severely inhibit product information exchange. (This problem is discussed in detail in [Sects. 2](#) and [3](#)). In this section, we will focus on the conversion of a design model to a manufacturing feature model using feature technology. A new trend of CAD/CAPP/CAM/CNC application integration is introduced in detail as well.

At present, most commercial CAD software tools support both solid modeling and feature-based modeling. A product model is usually constructed with the convenience of geometric construction, with features available in the CAD packages. When manufacturing processes are to be defined with the existing CAD models, the challenge of data reuse occurs. The majority of CAM tools on the market are feature-based, and certain specifically defined manufacturing features have to be used to define the processes that enable associativity with cutters, machine tools, jigs and fixtures, tool paths, and process conditions.

FR was the first challenge for feature applications in a domain based on CAD technologies, even though so far significant progress has been made. When a product is modeled simply in solids, FR is used to acquire engineering semantic features, such as manufacturing features, from CAD models. When the CAD model is created using a hybrid method of solid modeling and feature-based modeling, FR is still necessary to identify the manufacturing features. This is due to the fact that the design features will not be applicable in the manufacturing model because of the different definitions between design and manufacturing features and incomplete definition of geometry by design features. The interoperability problem between these design and manufacturing domains have been commonly recognized in industry due to historical CAD/CAM technology evolution. That is why FR plays a key role in feature-based CAM [34]. More elaboration follows in the next section.

However, inherited problems also exist in the FR approach, due to the restrictions of the hard-coded feature patterns to be recognized. To address the interoperability issue among computer-aided solutions at the feature level, other techniques have recently begun to emerge. In theory, with recent research progress in advanced feature-based modeling scenarios, if the CAD model is created with well-defined design features, there are two options. If the association between the CAD model and the manufacturing model is not required, then the FR approach can still be applied to generate manufacturing features from the resulting part solids. On the other hand, if the associativity is to be kept for updating future changes, then feature conversion is expected to map the design feature model to a manufacturing feature model. Feature conversion is also further discussed in [Sect. 5.2](#).

5.1 Feature Recognition

FR is an interpretation of a geometric model to identify features [10], and can be achieved by the user interactively or automatically by algorithms. With the user-driven approach, the user can select certain entities in the parametric model to define a feature [108]. For example, a user will pick three imaginary faces and two real faces to define a notch in the boundary representation (B-rep).

However, to realize complete CAD/CAM integration, automatic feature recognition (AFR) is necessary. AFR is the process of matching the parametric model with the predefined generic features to identify features. Babic et al. [1] specified three interrelated tasks which are necessary for AFR: geometric feature extraction, part representation formation suitable for form identification, and form feature matching. The specific tasks in this process are searching the database to match geometric patterns; extracting recognized features from the database; determining feature parameters; and completing the feature geometric model [77].

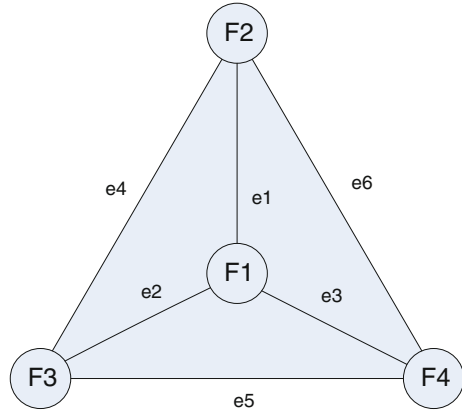
5.1.1 Rule-Based Methods

Rule-based methods) use production rules to depict features. The rules show the necessary conditions for the elements in the model such as convexity, perpendicularity, or adjacency. The expert system then uses these rules to perform the FR [10, 34]. Rule-based methods) were the initial ideas for FR. However, they have such obvious drawbacks that writing rules for all the features is a huge project and recognition will consequently be very slow.

5.1.2 Graph-Based Methods

A common data structure for B-rep models is the graph [10], especially the face-edge graph as shown in Fig. 7. Nodes represent the faces in the model and links of

Fig. 7 Face-edge graph for feature recognition



nodes represent the edges between the faces. The properties of the links represent the adjacency relations between the faces. In this way, the graph-matching method realizes FR [34]. Graph-based methods are currently the most frequently used FR technique, largely due to their efficiency. A variety of approaches with respect to each task are classified in Table 4.

A comparison between DBF and FR is illustrated in Table 5. Although great research effort has been spent in this field with a certain amount of progress, there are still some limitations associated with feature technology. Especially among computer-aided systems, a multiple view for engineers across different disciplines is required to support interoperability. To further extend the capability of feature technology in the enhancement of interoperability among different computer-aided systems, some new technologies (such as associative feature, unified feature,

Table 4 Classification of AFR approaches [1 CiI]

| Form feature extraction | | Pattern recognition |
|------------------------------|---|----------------------------|
| Geometric feature extraction | Form feature identification | |
| 1. External approach | 1. Syntactic pattern recognition | Logic rules |
| 2. Internal approach | 2. State transition diagrams and automata | |
| | 3. Logic (if-then) rules and expert systems | |
| | 4. Graph-based approach | |
| | 5. Convex-hull volumetric decomposition | |
| | 6. Cell-based volumetric decomposition | |
| | 7. Hint-based approach | |
| | 8. Hybrid approach | |
| | 1. Graph-based approach | Artificial neural networks |
| | 2. Face coding | |
| | 3. Contour-syntactic approach | |
| | 4. Volume decomposition | |

Table 5 Comparison between design by feature and feature recognition in manufacturing

| | Advantage | Disadvantage |
|--------------------------|---|---|
| Design by feature [77] | <ul style="list-style-type: none"> • Rich information associated with models • Real-time manufacturability evaluation • Concurrent design and process planning | <ul style="list-style-type: none"> • Restricted design creativity by the feature availability • Limited and specific features • Complex and undefined features resulting from feature interactions |
| Feature recognition [53] | <ul style="list-style-type: none"> • No need for input feature information • No constraint to design creativity • Possible automatic recognition | <ul style="list-style-type: none"> • Complex recognition algorithms • Limited features to be recognized • Input required for associated manufacturing information |

multiple view feature, and semantic feature modeling) have been introduced, all of which are covered in the scope of semantic modeling [8, 11, 19, 61].

5.2 Feature Conversion

Feature-based design is a relatively new approach for CAD/CAM integration [74]. For FR, it is a process that transfers low-level parametric models into high-level features. If we set up the product model with features initially, the extraction of features will be quite easy. There are several requirements for a feature modeling system, as follows:

1. The system must be interactive and graphical, as this is the best way to support the modeling system.
2. There must be a library for the storage of generic descriptions of features, and a mechanism to create instances of features by specifying the features.
3. Constraints must be represented and maintained consistently to guarantee the validity of features.

Design features usually consist of form features coupled with functions, design intents, and other design-related information. As mentioned above, manufacturing features consist of special form features coupled with distinctive machining operations and other manufacturing-related information [29, 30]. Such different feature domains are supposed to be associative, to cater to constant change throughout the product lifecycle; hence, after design modeling, there is a need to convert the CAD feature model into a CAM feature model. Much research has been dedicated to this conversion process [74], which can be divided into three parts: form feature mapping, dimension mapping, and the mapping of other

attributes such as tolerance or surface finish; more details can also be found in the work of Gao et al. [29].

5.3 Feature Interaction

Most research in this area has focused on simple FR and conversion, while feature interaction is often encountered in practice and causes difficulties in FR and conversion. Some researchers have tried to recognize composite features as a combination of simple features, but have not resolved the issue completely [69, 94]. Lee [55] concludes that are based on nine kinds of simple features: step, blind step, slot, blind slot, pocket, hole, wedge, fillet, and sector, and brings out the projective FR algorithm to recognize composite features. Gao et al. [30] propose a mathematical description of the feature mapping process to solve the feature interaction problem. However, all these works can only partially address the problem, and feature interaction remains an extremely challenging research issue.

5.4 CAD/CAPP/CAM/CNC Integration

At the end of the twentieth century, most research effort had been put into the integration of CAD/CAPP/CAM with the neutral file standard and feature technology. However, for a complete manufacturing process chain, the critical interoperable connection problem from CAD to CNC is still not fully solved.

In traditional CNC manufacturing, control is based on axis-movement description programming techniques (G and M codes) [38] which cannot perfectly support the advancement of CNC machines. Hence, machine manufacturers add their proprietary instructions to the standard [105]. Consequently, there is a need for specific post-processing programs for different configurations of CAM tools and CNC machines, which is a big obstacle for the interoperability of CAM/CNC applications. Furthermore, CAM systems add manufacturing-related information to the design, such as machine processes, tools used, and operations. However, after post-processing, the output files are NC programs, which are only machine-interpretable by CNC machines. The result is that the data translation is a one-way process, involving huge information loss [99].

STEP greatly helps the CAD/CAPP/CAM integration process, and STEP-NC has been developed with the aim of better CAM/CNC integration. STEP-NC is a standard specifically for NC programming, helping to achieve the goal of a standardized CNC controller and NC code generation facility. This standard has two notable advantages: first, STEP-NC is vendor-independent: if the vendors accept this standard, then a neutral data is achieved for exchange; second, STEP-NC files have the data regarding “what to do” instead of “how to do,” which are easily accepted by different intelligent CNC controllers [99]. There are three types

of STEP-compliant CNC: (1) conventional control; (2) new control; and (3) intelligent control [92, 105].

For conventional control, STEP-NC translators read the STEP-NC file and output an NC file, which is similar to post processing. It has achieved partial interoperability, as different configurations can be connected using the same neutral file. With this method, the CNC machines do not need to be retrofitted [92]. The new CNC controllers can process the STEP-NC file inside the CNC machine and then convert it into NC programs which consist of G and M codes. At present, this method does not have many intelligent functions, and most researchers are building the CAM/CNC integration in this way [91, 106]. Xu and Wang [104] proposed a G code-free machining procedure. In their system, the CNC controller will make full use of the information stored in the STEP-NC file, such as the work plan, work step, machining features, and cutting tools, to work out the NC codes using its own programmable control language.

Intelligent control is the most promising type for STEP-compliant CNC. As both design and process plan information are stored in the STEP-NC file, many intelligent functions can be achieved by the CNC controller [105]. Suh et al. [91, 92] have developed a new CNC controller called STEP-CNC. It uses the STEP-NC file as input and can realize intelligent machining control functions, such as decision making for unexpected changes, program validation at the time of execution, monitoring, and recovering.

6 CAD/CAE Integration

CAD systems are commonly used for modeling the geometry of a product with a variety of tools; CAD geometry is used as input for FEA in CAE. CAD and CAE data models are always different from one another, due to the nature of the operations they carry out. In order to decrease the length of the product development cycle, the integration of CAD and CAE is in high demand; numerous efforts have been made in recent decades. Ideally, the integration of CAD and CAE will efficiently decrease the design cycle time, reduce cost, and simplify the fine-tuning process for the product; Gabbert and Wehner did a feasibility study on CAD/FEA integration as early as 1993 [28], and many researchers continue to work toward a seamless integration of CAD and CAE systems, without yet to achieving a satisfactory result. Gordon [31] summarized CAD/CAE integration into three approaches: geometry conversion, CAD-centric geometric modeling, and CAE-centric geometric modeling.

In the geometry conversion approach, CAD geometry is used and then converted into simulation mesh geometry. In this approach, the same geometric source is used in both design and analysis. However, this type of integration can only be used with simple parts, such as pipelines.

The second approach uses a CAD-centric geometric model. In this type of integration, the CAD solid models contain too many details that are not suitable for

the CAE-required abstracted models. There thus needs to be an idealization process that includes detail removal and dimensional reduction. Currently, the idealization process is a major obstacle for CAD/CAE practice. However, due to the ease of access to the modern 3D feature-based CAD technology, researchers tend to use this type of integration.

The third approach can be classified as a CAE-centric geometric model. The simulation model is built first, and is based on the design concept and analysis method. After analysis, verification, and modification of the simulation feature model, designers are to work out full details and manufacturing features to support downstream process planning. This type of integration is recommended by Gordon but requires analysts to know upfront about the product's function details.

The subsections below are intended to introduce the key technologies and remaining problems related to integration. First, data interoperability issues between CAD and CAE are discussed, followed by an introduction to geometry transformation practice. Recent feature-based integration research is also reviewed in detail. The basic concepts of three specific methods—the multimodeling method, common data model method, and analysis feature method—are introduced along with the visual framework structures. The benefits, technological improvements, and limitations of these three methods are discussed as well. Though the feature-based product method for CAD/CAE integration is in development, there are still some gaps to be filled.

6.1 Data Interoperability Between CAD and CAE Systems

Many researchers have tried to build an integrated CAD/CAE data model. Re-mondini et al. [73] developed a unified data model supporting both design and structure analysis activities, which can build the bridge between the CAD model and CAE analysis. However, this model is restricted to the treatment of linear analysis. For interoperability, Foucault et al. [26] recommended using a polyhedral model as a transitional model between CAD and the finite element method (FEM). However, this method has to modify and update the product design model repeatedly during the product development evolution.

Semantically, data is the lowest level of information, which is used in software to represent different kinds of information in product design. STEP standard AP 209 provides a means to build an integrated model, including nominal geometry (CAD), various idealized CAE geometries, and associated FEM analysis models and results, along with PDM and separate version control [48]. Users can customize the combination as needed. Liang et al. [58] raise the idea of using an integrated product data-sharing environment (IPDE) based on STEP, to allow CAD/CAM/CAE programs from different vendors to share data conveniently. This model is mainly based on several STEP application protocols: AP 203, 209, 214, and 224. Among these, AP 203 specifies data structure definitions for the configuration-controlled 3D designs of mechanical parts and assemblies, while AP

209 supports the design elements through analysis of composite and metallic structure. However, implementation of STEP in CAD/CAE integration is still limited, because STEP has mainly been developed for CAD/CAM integration.

6.2 Geometry Transformation for CAD/CAE Integration

A common platform that can contain information from both the CAD and CAE sides has been a popular research topic since the early 1980s. Early research mainly focused on the idealization of CAD models and automatic mesh generation. A workflow of the CAD/CAE integrated modeling method has been suggested by Li et al. [57], as shown in Fig. 8. CAD models are usually set up to satisfy requirements for design, process planning, and manufacturing [23] and therefore usually contain too many complex details for CAE analysis. Hence, the models need to be idealized before they can be subjected to CAE application. In most cases, the idealization process will not affect the accuracy of the analysis but can reduce analysis time significantly. There are two main methods for CAD model idealization: CAD detail feature simplification and dimension reduction [5, 23].

6.2.1 CAD Detail Feature Simplification

Detail simplification is the process of removing those unnecessary detail design features that do not affect analytic accuracy or mesh quality but do increase analysis time. Detailed features refer to the small shape features of the product model, such as small fillets and minor local ribs. Usually, these small features have little influence on the analysis result compared with the overall parameters of the model. Ji et al. [49] group the detailed features needing to be removed into a number of types, including chamfer features, edge blending features, thread features, groove features, holes features, pad and boss features, and slot features.

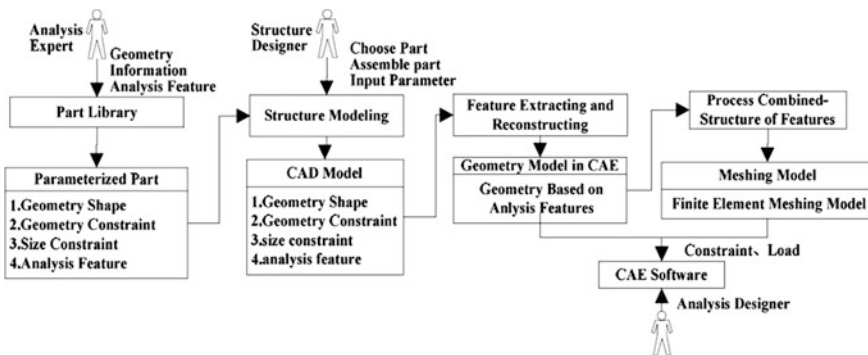


Fig. 8 CAD/CAE integrated modeling scheme [57]

To remove these features automatically, a simplification processing module must be built, for which functions include unwanted detail FR, selection, removal, and recovery. The process involves three steps: searching all the features in the model, determining the parametric information of the features by feature recognition, and carrying out rule-based simplification.

6.2.2 Dimension Reduction

In CAE systems, wireframes are used to represent beams and sheets for plates and shells. This kind of representation requires a process of abstraction. The mid-surface approach is suggested for this abstraction [24, 72], which usually involves three steps:

- *Face judgment.* Judge whether the two planes of the model can be faces. If yes, then the mid-surface is created between the faces.
- *Mid-surface modification.* This step modifies the mid-surfaces by removing those small facets that have little effect on the analysis results.
- *Extend and seam.* The mid-surface model is completed as a whole with extension and seaming of faces.

Currently, the idealization process still requires human interaction, while automatic mesh generation is realized by most commercial CAE tools [23, 27].

6.3 Feature-Based CAD/CAE Integration

As mentioned above, all three of the methods proposed by Gordon [31] require two separate models for one product, to the severe detriment of efficiency. Moreover, there are only geometric connections between these two models. The connections are thus a one-way process, and some semantic information needed for CAE analysis is lost [23]. Recently, researchers have been trying to develop a unified data model and concurrent modeling environment for seamless CAD/CAE integration.

Kao et al. [50] recognized that most of the feature models can be used in both CAD and CAE software, including geometric and non-geometric features. In their work, the changes propagated from CAD model to CAE model appear to be automatic. In fact, all the related CAD parameters had to be recalculated beforehand and exported into a spreadsheet, and the corresponding changes in the CAE model had to be made interactively. Chen et al. [16, 17] discuss semantics in information entities, relations, and constraints in each phase, and generalized common entities in order to develop a consistent product information model. In the course of their work, they then created a conceptual framework by applying the unified feature concept for CAD and CAE model integration [17].

Features, a form of well-defined data structure expressing engineering patterns associated to geometric entities and relations, are recognized as the basic and essential entities for product model design and interoperability between different types of software, such as CAD and CAE software. However, the feature association of design models between CAD and CAE is considered to be the main area of difficulty in terms of integration. For example, design form features used in CAD are usually represented geometrically, while features used in FEM have mesh and material data, which are derived from the imported CAD geometry without considering design features. These different types of features are easily confounded can cause mistakes in the design updating phase.

Multi-model technology (MMT) introduces object-oriented technology (OO) into the product modeling process and combines OO with feature-based modeling technology. It utilizes OO to create the object model of a product and uses feature-based modeling technology to build the model of an object. In this way, it can sustain system-level modeling along with design-level modeling. Because the object model consists of multi-models, it is called MMT [107]. The object model of the product is a multi-model structure (MMS), consisting of a finished part model level (assembly model level), a rough part model level (part model level), a function model level, and a basic model level [107]. Every model in the MMS is created by feature-based technology in the design process. With the help of MMS, CAD engineers and CAE engineers are expected to work concurrently, and the integration of CAD and CAE can be achieved. Figure 9 shows a visual structure of CAD and CAE interaction processes.

Lee [56] contributed to a feature-based multi-resolution and multi-abstraction modeling approach. This technology is realized using techniques such as

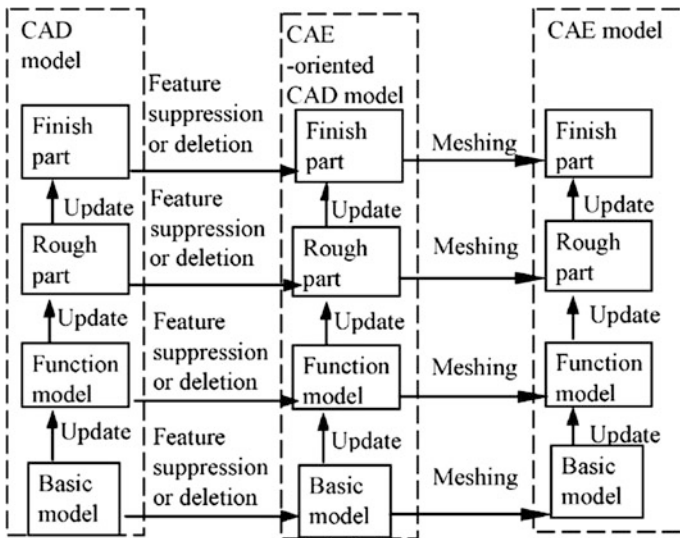


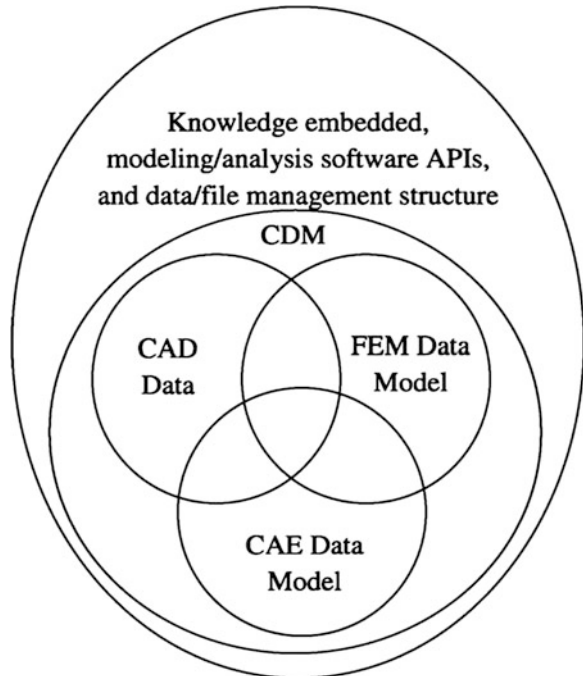
Fig. 9 The integration of CAD/CAE in MMT [107]

design-by-feature, non-manifold topological (NMT) modeling, multi-resolution solid modeling, and multi-abstraction NMT modeling. The CAD and FEA models are built up simultaneously into a unified master model, in which design and analysis features are embedded; this research supports the buildup of CAD and FEA models at multiple levels. There is a drawback to this method, however: boundary conditions such as load and displacement conditions cannot be transferred from CAD models into CAE models automatically.

With feature-based technology, if a product needs to be modified, a synchronization mechanism can be developed to update the FEM model with persistent connections between the CAD model and the CAE model. This could mean significant benefits for the product development process. However, this feature-based approach also requires higher knowledge and skill competency for the analysis engineers, who must initially extract useful analysis features from the CAD model to create the associated CAE model. The CAE engineer needs to give specific working condition definitions for such analysis features in the early stage.

Following the development of feature-based methodology, Gujarathi and Ma [33] tried to integrate CAD and CAE models using a joint data structure called a common data model (CDM). This CDM consists of semantic design parameters used in three ways: building the CAD model, building the FE analysis mesh model, and performing engineering analysis functions with the assistance of knowledge-based algorithms and software APIs. Figure 10 shows the basic concept of CDM [33].

Fig. 10 General working aspects of CDM [33]



In the proposed method, the CDM is initially generated by an engineering concept calculation module which works out the key driving parameters within the engineering project scope. The CDM is used to store the initial conditions and final results of the engineering conceptualization result in parameters and their constraints. In this conceptualization procedure, basic physics/chemical principles are implemented and verified.

A structural designer then constructed the CAD models by retrieving templates from a part and assembly library; the templates provided input for design parameters, and defined the assembly relations among parts.

Third, product's FEM geometry information, based on the analysis feature information that was already embedded in the CAD templates, was constructed automatically in the CAE system. Since 1990s, a feature-centric CAD/CAE integration approach has been developed by a number of researchers [56, 57, 82, 110]. In an early effort [82], a part library with built-in analysis features was first established by an expert CAE engineer. In the proposed method [32, 33], geometrical CAE meshes are generated in sequence using an automatic meshing technique. The meshes of features are ultimately combined into a complete mesh model for the product, which is guaranteed by a structure-combining algorithm. Finally, CAE analysis is automatically carried out and the results are also recorded in the CDM.

This method offers centralized design parameters and data for CAD/CAE. The CDM method for CAD/CAE integration has two specific advantages. First, it supports parametric design and analysis in the integrated CAD/CAE environment. Second, CDM updates its parametric data dynamically over the processes involved in design consolidation. The content parameters in CDM largely belong to three general categories: geometric, non-geometric/functional, and intermediate design parameters. The structure of the CDM is shown in Fig. 11 [32].

The approach can also be extended to include engineering rules used in various models. These rules can then be consistently applied to multiple domains. Such a centralized "control board" enables an enhanced control mechanism that offers the flexibility of adopting and changing a variety of design codes, standards, and expertise in the cycle of design procedures. As a kind of parametric data model, CDM data can also be further customized to incorporate manufacturing requirements.

Although quite flexible, the proposed method has some limitations. The most difficult task in the initial phase of parametric CAD modeling is the associative relation model development. The initial identification of parameters and the logics of different kinds of relationships require considerable programming and set-up effort for automatic model generation and updating.

Further, the design procedure in the terms of computer system operations has to be fully defined beforehand in consideration of building the CAD and CAE analysis models with logical constraints. Therefore, the proposed method offers long-term efficiency only for those well-established, generic, and set design problems [33]. Efficiently or expeditiously conducting simulation for the testing of candidate solutions and demonstrating design scenarios for provisional customers

| | | | |
|-----------------------|--|---|--|
| Associative relations | Project Management Data | | Process flow control parameters and interfaces |
| | Common Data Model | | |
| | Customer Requirements R1, R2, R3,...Rn Operational parameters | Design Specifications S1, S2, S3,... Sn Governing codes & standards | |
| | Conceptual design & engineering parameters and constraints, C1, C2, C3,...Cn | | |
| | Conceptual (e.g. mid-plane) CAD modeling parameters and attributes | Conceptual CAE analysis setting parameters and intermediate results | |
| | Detailed design & engineering parameters and constraints, D1, D2, D3,... Dn | | |
| | Detailed (e.g. solid) CAD modeling parameters and attributes | Detailed CAE analysis setting parameters and intermediate results | |
| | Versioned final design parameters and constraints | | |
| | Final CAD modeling parameters, attributes and drawing data | Final CAE analysis setting parameters and intermediate results | |

Fig. 11 Structure of CDM with the progress of the design process [32]

can be problematic because of the ad hoc procedures involved and the short response time required.

The authors believe the CDM integration approach [32, 33] has two contributions. First, the method abandons an oft-required feature reformation process (from CAD to CAE) that is still technologically immature, and instead binds the analysis features within the parameterized CAD model [82, 110]. More progress has been made to improve the reliability of obtaining the analysis feature from the CAD model [57]. Second, automatic mesh generation based on analysis features is a straightforward technique. The meshing method improves the quality of the mesh model. The improved method leads to better analysis results and shorter simulation time. In addition, more complicated component shapes and assemblies can be managed.

This common data model involves only the preliminary design and is limited to sizing and the essential operational concepts. Refinement of the model by adding more design features has to be carried out before it can be usable in day-to-day industrial applications.

7 Toward Feature-Based Integration and Interoperability in Chemical Process Engineering

The complexity of chemical process design and engineering requires engineers to work collaboratively across disciplines. Existing software tools have allowed engineers from disparate domains to deal with the complexity embedded in each specific domain. However, interoperability of the heterogeneous data generated by different tools remains a problem [62]. This reality highlights the urgent need to integrate the software tools involved in any given chemical process project to enhance interoperability, not only on the levels of syntax and structure, but also on the semantic level. As should be clear from the technology reviews in previous chapters, semantic modeling and feature technology have been adopted by researchers to construct a variety of integration frameworks. This section proposes one such framework, under which the semantic feature associations between two domains are analyzed and a new, more efficient design process is proposed based on the concept of collaborative engineering. A case study further demonstrates how the framework functions, and allows engineers from different domains to work collaboratively within it.

7.1 Integrated System Architecture for Chemical Process Engineering

The common project engineering practice in chemical process development involves multiple disciplines, such as chemical process engineering and mechanical engineering. Ideally, the engineering design efforts should be coordinated coherently, with close interactions among relevant disciplines due to their heavy dependency on one another. Traditional discipline-centric engineering practice and the relevant engineering software tools are becoming outdated, because networked computer information technology has made interdisciplinary collaboration much easier. To address this kind of industrial demand, the authors have proposed a system integration architecture [103] based on a common framework of semantic modeling and feature technology, and consisting of disciplinary modules such as mechanical and process design modules. The centrally unified feature management system (a common base module) consists of a product feature module and a process feature module, which are built on top of a networked federation of data repositories representing different disciplinary domains.

The improvement of the proposed semantic integration framework over the individual disciplinary engineering approach is that it incorporates semantic interoperability. The feature information will be retrieved from the files or databases generated by different domain software tools and mapped onto a central database according to the relevant semantic schema and a generic mapping mechanism. Based on the data collected, the central unified feature management

system will generate a view according to domain-specific schema and display it to the domain-specific engineers through their respective user interfaces. An ontology library and a knowledge library have been developed to support semantic feature mapping. As shown in Fig. 12, Module 6, the central unified feature management is the core of the system, maintaining and validating all the features according to a unified scheme with the related mapping mechanism, and managing view generation and updates. In addition, a standard feature library can be established using the generic data feature structure specified by Xie et al. [103], which will facilitate semantic feature mapping and also reduce the modeling workload of mechanical engineers.

Due to space constraints, the framework provided here lists only a few of the software tools involved in chemical process engineering. There are more software tools used in industry, however, and these vary from company to company. Extensive software tools have been developed and applied in chemical process engineering as well. Within this framework, whenever a new version or a completely new software tool is created, only one new “translator” needs to be added into the shared interface library. This leads to a considerable decrease in the development efforts needed, as compared to merging different modeling schema into an integrated schema [4].

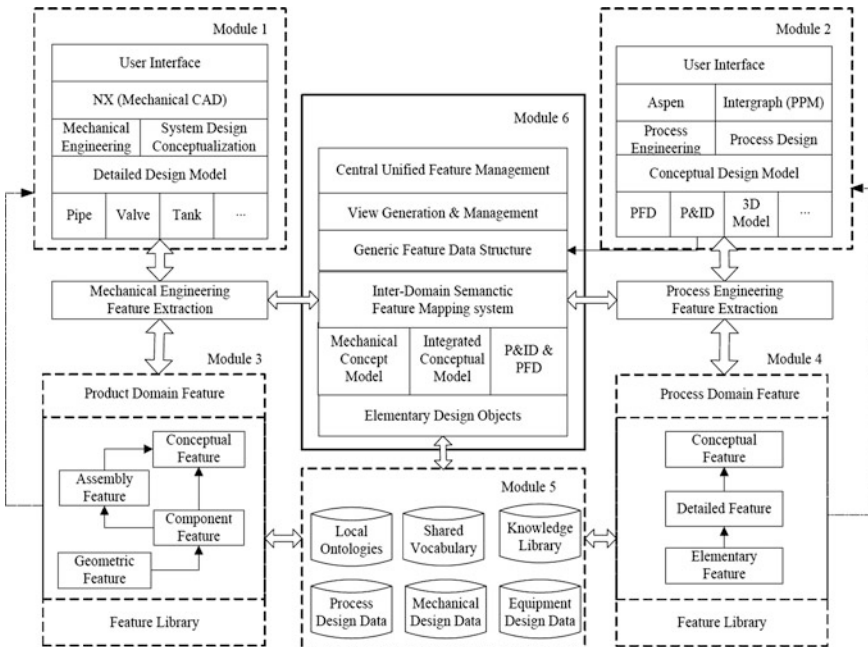


Fig. 12 Integrated system architecture [103]

7.2 Semantic Feature Associations Between Process Conceptual Design and Mechanical Detail Design

The first challenge in facilitating semantic integration in chemical process engineering is to identify the semantic feature association among the phases of the lifecycle [60]. The activities involved in chemical as well as mechanical process engineering and design lead to the generation of domain-specific features. These features are classified, as suggested by Han [35], into two categories: chemical process conceptual design features (CPCDFs) and mechanical detail design features (MDDFs), as shown in Fig. 13. CPCDFs are the features created in the chemical process conceptual design and engineering phases, which are designed to satisfy the requirements of the chemical process projects, as in, for example, the capacity of a chemical plant under construction. The parameters involved in CPCDFs can be mapped to the constraints, which will influence the downstream mechanical detail design [60]. MDDFs are the features created in the mechanical engineering and design phase to satisfy the requirements and be subject to the constraints stated in the process conceptual design. The specification of the equipment design will in turn place constraints on the process conceptual design. These mappings are implemented by knowledge-based reasoning, as shown in Fig. 13.

The semantic associations between feature parameters that are associated with equipment engineering and design are shown in Fig. 14. The corrosion allowance constraint (CAC), temperature (T), pressure (P), and residence time (RT) can be considered by the input and output parameters, which are specified based on the project requirements during the conceptualization phase [33]. Further, the shell thickness constraint (STC) is derived by T and P. The flow rate (Fr) is calculated based on the capacity (Cap) requirement and diameter of piping (DP), which will

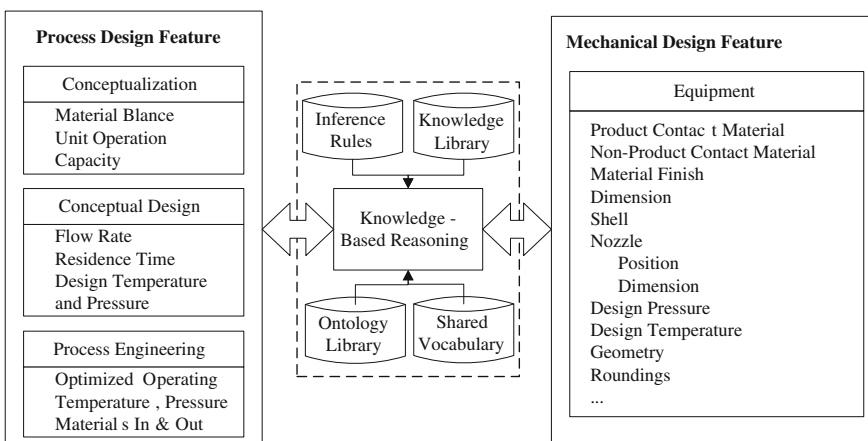


Fig. 13 Semantic feature associations between process design and mechanical design

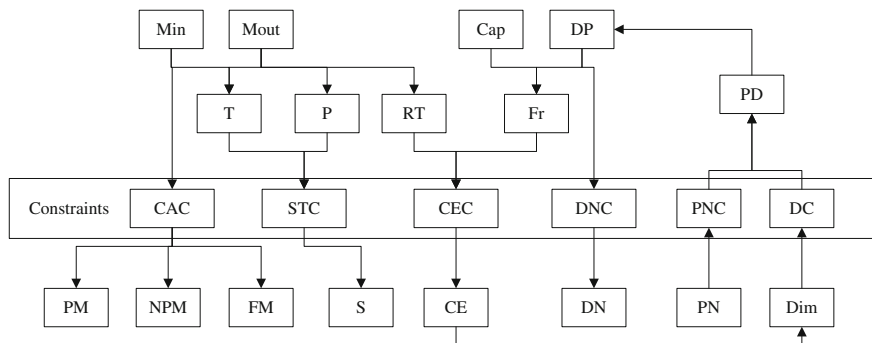


Fig. 14 Semantic associations of feature parameters and constraints

further determine the capacity-of-equipment constraint (CEC). And with DP alone, the diameter of nozzle constraint (DNC) is identified and will further determine that diameter of nozzle (DN) should be equal to the DNC. Meanwhile, the equipment's shell thickness (ST) and capacity-of-equipment (CE) should be larger than the STC and the CEC, respectively, while product contact material (PCM), non-product contact material (NPCM), and finish material (FM) are identified by the CAC. Further, mechanical engineers will work out the dimension (D) and geometry of the equipment, which determines the dimension constraint (DC). Similarly, DCs are used to describe the relationship between the position of nozzle (PN) and those position-of-nozzle constraints (PNC). Lastly, the DP is identified based on the piping design (PD), which is influenced by the DC and the PNC.

The mechanical design features associated with the chemical process conceptual design features should be kept consistent by implementing an active checking mechanism known as the “association” [33]. This has to be implemented in two ways. Each CPCDF and its properties are mapped to constraints specified for mechanical design, and any later changes in the CPCDF will be reflected in the update of the constraints, which will further influence the parameters in the MDDFs. Similarly, further design changes by mechanical engineers within MDDFs will update those related constraints that are mapped from the CPCDFs; then the constraint changes will trigger CPCDF updates. If there is any constraint conflict emerges during the updating process, the change will be hold up, and a report generated for engineers' review in order to determine the next step of the reasoning path.

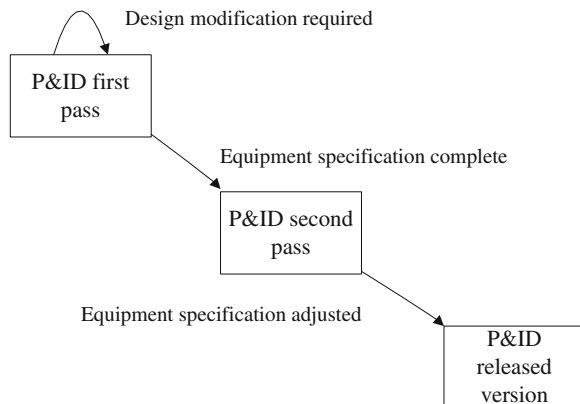
7.3 Proposed Workflow Under the Integrated Framework

Conventionally, chemical process design and mechanical design work are sequentially coupled with verbal consultations between engineers. After the process design engineer works out the process specifications, mechanical detail design

begins. However, this second stage is often delayed by several iterations of the process conceptual design modification. The specifications of the mechanical design may then require that the process conceptual design again be changed, especially when “non-standard” equipment is applied. However, changes to the process design will also lead to adjustments in the specifications of equipment and hence of the mechanical design [19]. Several iterations of both phases are usually needed before the process design and mechanical design are finalized. The work associated with these iterations is tedious, time-consuming, and error-prone. It is difficult to maintain consistency, as engineers’ work during iterations will often conflict with those constraints defined in earlier cycles, without the engineers noticing [4, 21]. An example is shown in Fig. 15 [4].

To reduce the amount of time spent on interdisciplinary collaboration during design phases, a new design process is proposed here by the authors based on collaborative engineering principles under an integrated framework, as shown in Fig. 16. For example, given a capacity expansion project in chemical engineering, the project scope and reusability of knowledge are first determined; the material balance and operation capacity are identified in the conceptualization based on the project requirements. Instead of working sequentially in the conventional design process, conceptual design, process engineering, and mechanical design are now implemented in a concurrent and collaborative environment. The associations involved are supported by a systematic knowledge-based reasoning procedure [101]. Meanwhile, engineering constraint checking should be implemented to keep designs originating from different domains consistent. If the design change is rejected by the constraint checking module, the engineer can retain the most recent valid model while trying another round of iterations to reach a new solution. In this collaborative work environment, the likelihood of redesign, as well as workload, is greatly reduced. Furthermore, knowledge will be extracted from the complete project case and added to the knowledge library for future reuse after the project is complete.

Fig. 15 Iteration of modifications in the conventional design process [4]



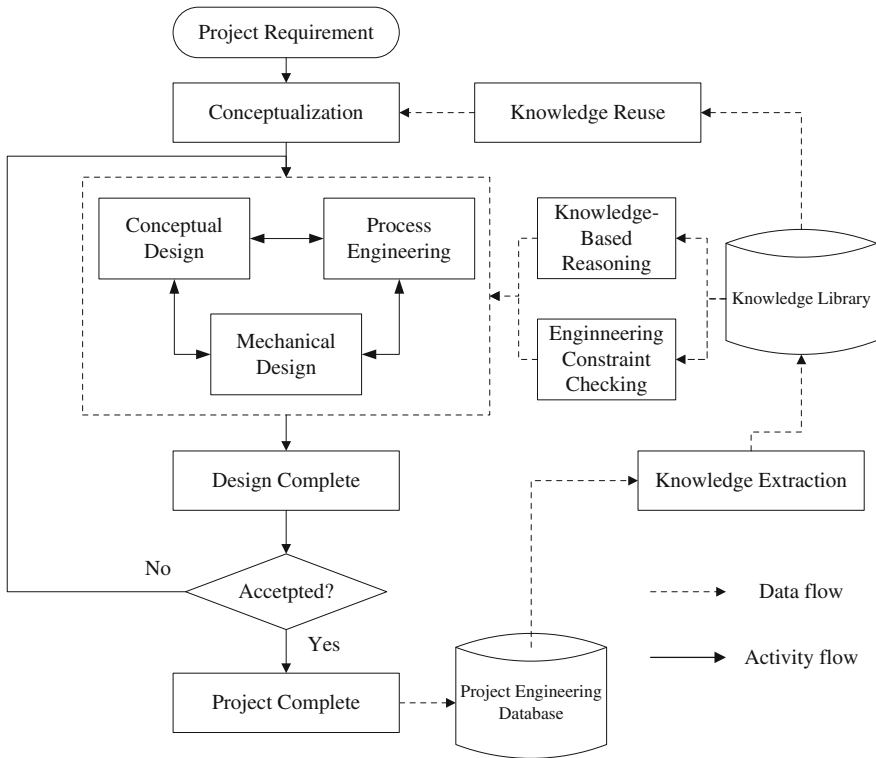


Fig. 16 Proposed design process flow under an integrated framework

7.4 Case Study

An example of the integrated system is shown in Fig. 17. Figure 17a shows the process and instrumentation diagram (P&ID) of the process based on its process flow diagram (PFD). Both the PFD and the P&ID are created with the conceptualization of design intent which can be expressed with a set of associated attributes. Such Attributes can be retrieved from a central database according to the project requirement analysis. The P&ID specifies the equipment, instrument, key piping, process control schema, and so on. For the downstream mechanical detail design, there is too much irrelevant and incomplete information, as only those equipment characteristics that are significant from the process point of view are specified. However, all of this information will be mapped onto the central database. With this information, as well as other information mapped from, for example, the PFD, the “standard” equipment can be selected from the equipment library, as shown in Fig. 17b. However, sometimes custom-designed equipment will be needed. In this case, the mechanical engineer can work on similar

equipment and just make some minor modifications. During this process, the knowledge library will provide data support.

However, there is again too much detailed and mechanically relevant information included in the solid model shown in Fig. 17b, such as small chamfering or filleting parameters embedded in the design features. Should the full solid model be transferred to process engineers or added to the process design model directly, it would cause a network burden with unnecessary information being transferred, and would also confuse the process engineers. Instead, a process engineering and design view, which has been tailored to include only the tank's process features, is generated according to the view definitions. This process view presents only process engineering feature properties, such as operating pressure, capacity, key dimensions, and other process-related attributes, and they will be further referenced in the process model as external data resources. Thus, the tank generated in Siemens NX shown in Fig. 17b is mapped to the tank in the pink wire frame in SmartPlant 3D, as shown in Fig. 17c.

8 System Architecture for Interoperable Network-Based Engineering Systems

Competition in global marketing forces companies to develop products in the shortest time with the highest quality. Collaborative engineering aims to shorten the length of the product development process. In collaborative engineering, tasks can be performed by engineers who are both spatially and temporally distributed. Two critical technologies help to realize collaborative engineering: web technology and agent technology.

8.1 Web Technology

Web technology enables centralized information integration through a shared web server and a central database [81]. It usually uses the client-server architecture to realize communication between those servers and distributed developing teams. Ideally, the collaborative engineering system will be web-based, semantics-enabled, comprehensive, and agent-based [36, 81].

Critical issues about web-based engineering systems need to be tackled. Expert systems tend to use a variety of software tools and computer systems. The system should therefore have the ability to support heterogeneous computer applications and data sharing. Distributed object technology such as the Common object request broker architecture (CORBA) and DCOM/ActiveX can solve these problems [68, 83]; the details of data exchange have been discussed in earlier sections.

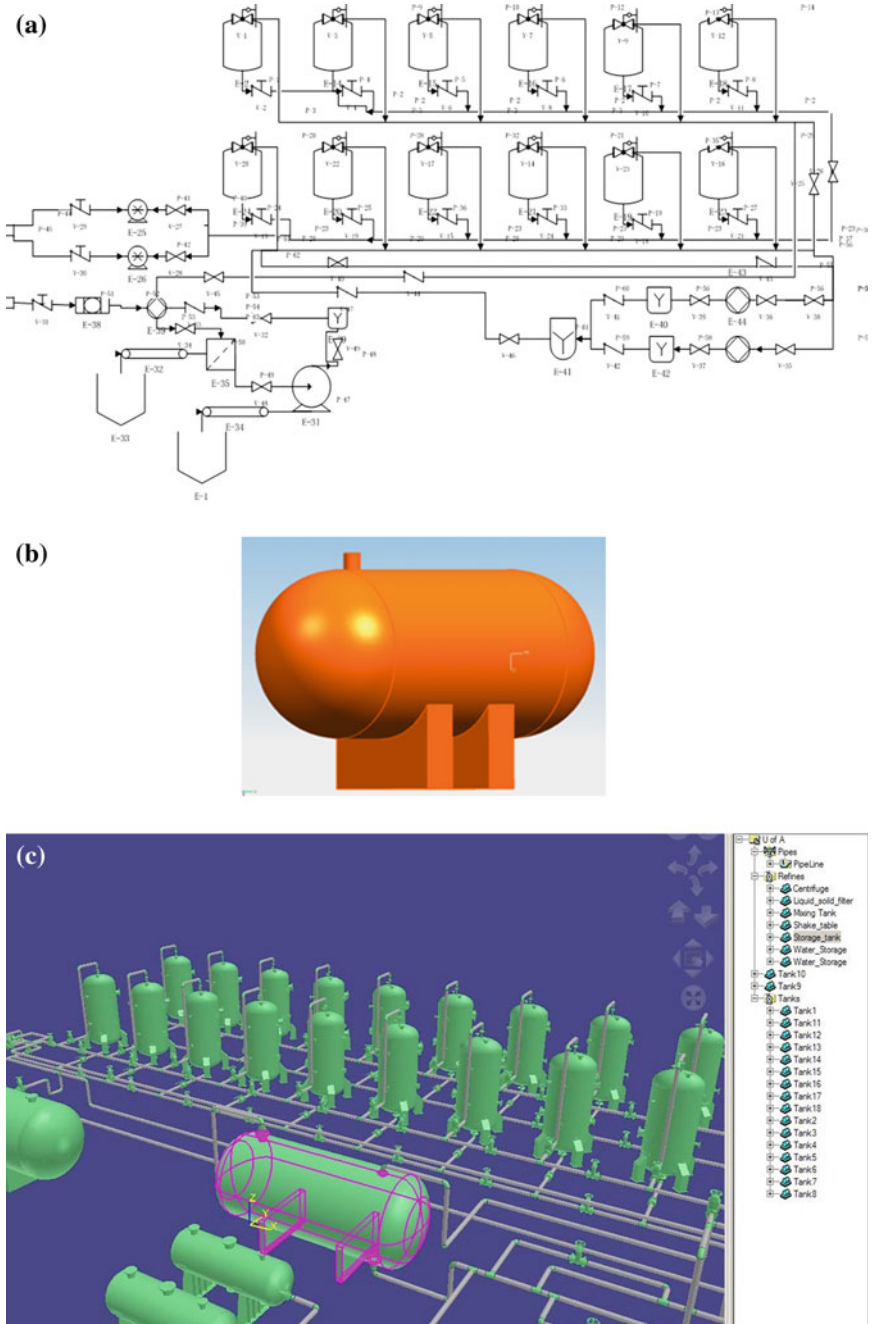


Fig. 17 An integrated system: **a** P&ID, **b** the 3D solid model generated in Siemens NX, and **c** the 3D model generated in SmartPlant

For collaborative design, multiple teams often work with the same model for different disciplinary purposes; hence, conflicts occur frequently. There has to be some rules for decision making. For instance, the system should notify those engineers in charge about the conflicting constraints and to the engineers make decisions via negotiation to resolve conflicts; sometimes multiple solutions exist simultaneously.

Web technology can only satisfy the data communication and exchange requirements of collaborative engineering systems. There are interwoven intellectual exchanges of opinion, consultations, and compromises that need complete, accurate, and sustainable information models instead of web technology alone. Further, the product-related data should be complete and can be translated into different application models; designers must have access to the complete design model if required to visualize, manipulate, and retrieve all the geometry and the semantics of the design, and negotiate modifications. It is also better for the collaborative engineering system to have flexible and modular architecture, and agent technology is useful in facilitating automatic process flow management requirements. Thus, ideally, the collaborative engineering system will be web-based, semantics-enabled, comprehensive, and agent-based [36, 81].

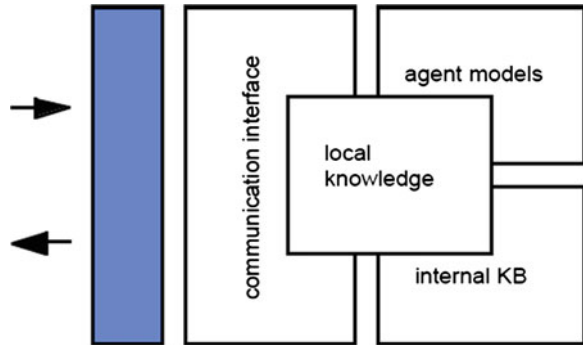
8.2 Agent Technology

Agents are programs acting for a user or another program under predefined conditions. The aim of Agent technology is to integrate heterogeneous, distributed, and semiautonomous knowledge-based software tools into a collaborative application [54]. There have been numerous efforts to develop agent-based collaborative engineering systems. Palo Alto collaborative test bed (PACT) is one of the earliest CE web-oriented platforms satisfying multiple sites and various disciplines. PACT is agent-based and allows agents working on different aspects of design to share and exchange information with one another [22]. Agent interaction relies on three elements [22]: shared concepts and terminology for communicating knowledge across disciplines, an interlingua for transferring knowledge among agents, and a communication and control language that enables agents to request information and services. Shen and Barthes [80] have developed a prototype of a distributed intelligent design environment (DIDE) in which the internal structure of an agent and the inter-agent communication mechanism are illustrated in detail. The internal structure of an agent is shown in Fig. 18.

8.3 Multi-Agent Systems

A web-based interoperable engineering system is composed of various engineering software tools that rely on different principles. In such systems, multi-agent

Fig. 18 Internal structure of an agent [80]



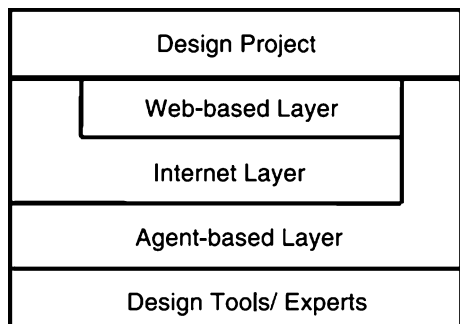
technology is more suitable and makes the systems more flexible. As each agent is coupled with a certain function and a well-defined application program interface, the engineering system can change its configuration based on practical requirements. At the same time, agent modules can be reused in different systems [54].

Wang et al. [100] have developed a distributed multidisciplinary design optimization (MDO) environment that supports seamless interaction between designers, agents, and servers. The architectural framework for MDO environments is shown in Fig. 19. Hao et al. [36] have developed a lightweight agent framework for mechanical product design by applying intelligent software agents, Web, workflow, and database technologies. The framework developed is compliant to Foundation for Intelligent Physical Agents (FIPA) standard, called autonomous agent development environment (AADE).

8.4 System Architecture

For the agent- and web-based interoperable engineering system, the target is to use software agents to reduce reliance on large, complex, centralized systems and to efficiently facilitate collaboration.

Fig. 19 Architectural framework of integrating the internet, web, and agent technologies for MDP environments [100]



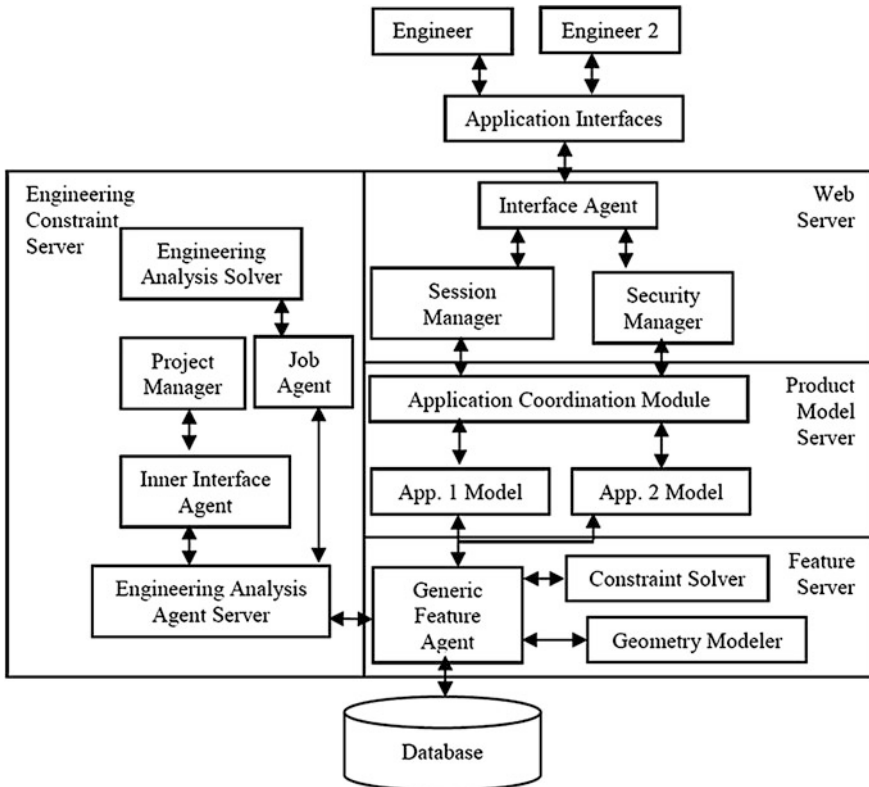


Fig. 20 A system design for agent-based engineering collaborations

Ulieru et al. [95] describe three layers of the system architecture: a low-level inter-networking communication support layer, a coordination layer (managing inter-agent cooperation through intelligent conversation/communication mechanisms), and an agent layer consisting of five categories of agents: interface, collaboration, knowledge management, application, and resource agents.

In this section, a unified feature model is applied as the system modeling basis to realize interoperability. A simplified architecture is proposed, as shown in Fig. 20.

8.4.1 Web Server

The web server contains an interface agent (IA), security manager, and session manager. The IA provides shared access for multiple users. It can instantiate different data for different users according to the users' requirements. The security manager is used to check whether the user has the right to access the product

model data, and what kind of access it is. Users are separated into several groups, each with different access rights. All management data are stored in the database.

The system supports shared access for multiple users, so a comprehensive data management system is designed for maximum concurrent access to the data. Access to data is mainly managed by the session manager. If multiple users with different priorities require the same data, the user with highest priority will get access to the data while other users have to wait (i.e., can only view the data) until his or her session is finished. If multiple users with the same priority require the same data, it follows the “first come, first served” principle: other users remain on a waiting list and can only view the data [9, 93].

8.4.2 Agent-Based Design

The inner IA is a bridge between the project manager and other agents. The project manager assigns new jobs and manages on-going job progress through the inner IA.

The engineering server agent (ESA) is the brain of the Agent-based design system. It communicates with the job manager to accept jobs and manages messages, then manages the data flow to operate the system. Its functions include transferring data files to and from the database, assigning jobs to job agents, and validating the finished design.

The job agent (JA) is responsible for automatic task arrangement. In the Agent-based design system, a design job is composed of job ID, task ID (which is formed following the task sequence), job parameters, and task files. When a new job enters the JA, the JA can automatically distribute tasks to the appropriate available designers following the task sequence.

8.4.3 Working Procedure for Agent-Based Design

1. The project manager gets access to the inner IA to give a new job to the system.
2. The ESA gets the message from the project manager and starts the function “starting job#.” It then reads the job data from the database and transfers it to the JA.
3. The JA receives the imported data and assigns specific tasks to various designers. Tasks will be arranged following predefined sequences.
4. When a job is finished, the finished design files are sent back to the ESA. The JA sends the design to the problem solver to be validated. If there is no failure, the design data is stored in the database and the job status changes to “finishing-job#.” However, if there is a defect, the process will go back to step 3 [36].

8.4.4 Downstream Application Management

To realize collaborative engineering, the collaborative design system needs support from many collaborating functional modules, which can provide services through agents as well. For example, features are managed by a feature agent. Other downstream applications can also be consolidated by an administrative downstream application management agent for their services or interactions.

The feature agent can provide feature objects for application packages and separate application packages for discrete users, so that users can use specific feature models for certain downstream applications. The feature agent has the functions of FR, feature extraction, and feature modification. The feature agent can also receive feedback from users and process it. Every time the feature agent modifies the feature model, it will call the constraint solver and the geometry modeler to validate the modified feature model. The constraint solver can check the validation of all constraints, which are part of the feature definition. The geometry modeler can validate feature geometry. Finally, the unified feature model in the database will be updated.

8.4.5 Database

The database provides physical storage for all kinds of data, including product model data and security management data. Geometric data and the unified feature models are stored within the database. The unified feature models are composed of various generic feature models, which are stored as data elements across tables [18, 19]. In this manner, the database manager can reorganize these data elements for flexible use by different applications [93].

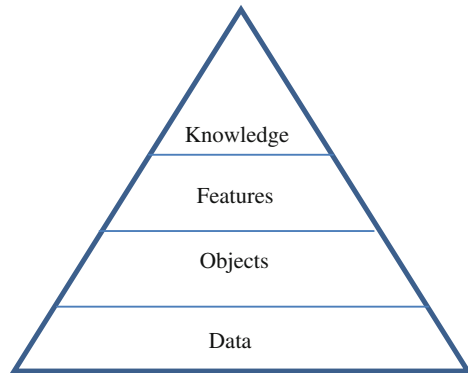
9 Information Views, Granularity, and Knowledge-Driven Engineering

9.1 Information Granularity

Granularity is the extent to which information is broken down into small components of computer system entities. Coarse-grained information model consists of fewer and larger components than fine-grained model. Granularity becomes an important issue for data modeling when trying to represent levels of information with data structures across systems or databases [25]. In practice, information can be granulated into four parts, as shown in Fig. 21.

Data. The most-granulated information type is data. Data as an abstract concept can be viewed as the lowest level of abstraction from which information and then

Fig. 21 Information granularity



knowledge are derived. Data on its own carries no meaning. For data to become information, it must be interpreted and take on a meaning.

Object. Objects can be thought of as wrapping their data within a set of functions designed to ensure that the data are used appropriately, and to assist in that use. The object's methods will typically include checks and safeguards that are specific to the types of data the object contains. An object can also offer simple-to-use, standardized methods for performing particular operations on its data.

Feature. A generic feature representation in a database can be expressed as shown in Fig. 22 [93]. A feature has *feature_id*, *product_id*, and *domain* as its attributes. The *feature_id* attribute is an object identifier, which can uniquely identify a feature object in database. *Product_id* specifies which product a particular feature belongs to. *Domain* is a predefined data type, which can be instantiated for *design*, *manufacturing*, or *analysis*, and their relevant setting parameters are stored in a domain table. A *feature* will also contain a list of referenced entities, a list of constraints, and a list of parameters. Dimensions and tolerances are regarded as subtypes of *constraint* bounded to certain geometrical entities.

Knowledge. Knowledge-based engineering (KBE) is OO and rule-based, where knowledge is the object and rules are the operations. Features can be a part of the knowledge, and rules are responsible for the reasoning and mapping of the features.

9.2 Information View

An information view is a selective set of information that is specially filtered for a purpose. As shown in Fig. 23, the functional view can be designed or implemented with a specific purpose and scope as well.

The design of user-specific and need-based information views plays a significant role in the integration of CAX applications. These views are context-dependent interpretations of self-contained subsets of information about the entire product model (EPM). With STEP technology, all of the functional views can be expressed

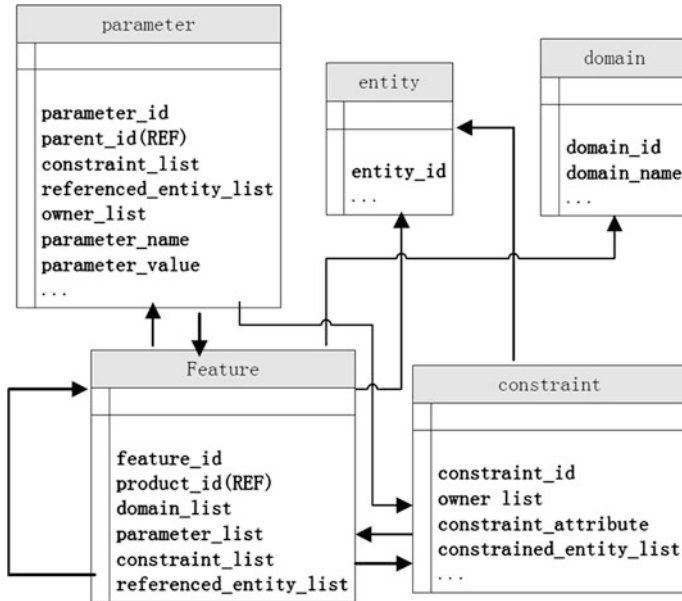


Fig. 22 Generic feature representation in a database [93]

with the same language, EXPRESS, and an arbitrary view can easily be translated into other views. Building a common product model representation is crucial to achieving different functional views. EPM describes information across applications, and contains the domain classification ontology and metadata. In practice, application feature sub-models can provide specific views of the EPM [93].

9.3 Introduction to Knowledge-Based Engineering

Knowledge-based engineering (KBE) is a special type of knowledge-based system (KBS) that focuses on product development activities such as design, analysis, process planning, and manufacturing. Stokes [89] defines KBE as “the use of advanced software techniques to capture and re-use product and process knowledge in an integrated way.”

There are many advantages of KBE. With the product and process knowledge stored in the system, KBE can reduce the time spent on routine work and also save time for innovation. As the expertise is stored in the database, companies will be less affected staff turnover. A major drawback, however, is that it takes time to develop and update the KBE system.

KBE is currently widely used in industry. KBE systems are usually developed by individual companies to generate product concepts using captured product and process knowledge, and can later be used to help prepare for FEA [12, 15].

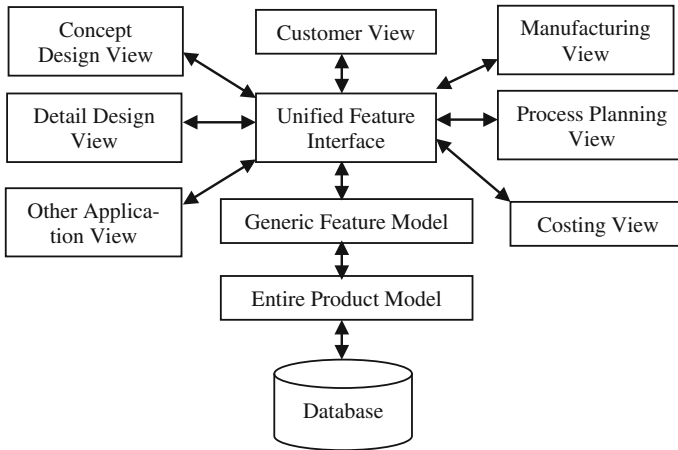


Fig. 23 Functional information views

A recent challenge is to integrate manufacturing-related knowledge into KBE systems to aid in manufacturing; this often takes the form of assessment of manufacturability and process planning [6, 85].

9.4 Foundations of KBE

Knowledge is central to KBE, and it interacts with information about how a certain product should be developed. Considering specific knowledge-based product engineering applications, knowledge can be managed with reference to three categories of information: geometry, configurations, and engineering rules. The rules are complex and powerful expressions composed with mathematical formulae and conditional statements [59, 75].

Geometry. Most product-oriented KBE systems have limited capability of CAD functions and hence are usually integrated with CAD packages. Very often, the output of the KBE system is a CAD model.

Configuration. This refers to a functional product model that is an assembly of a set of existing modular components. At present, vast KBE applications for configuration design are used with many real cases. A typical example is Toyota's configuration management system [88].

Engineering rules. This refers to specific domain-related or analysis-related knowledge that consists of well-organized logical rules and assists decision making based on the input conditions of engineering constraints imposed in product development processes. Figure 24 shows the framework of the KBE system.

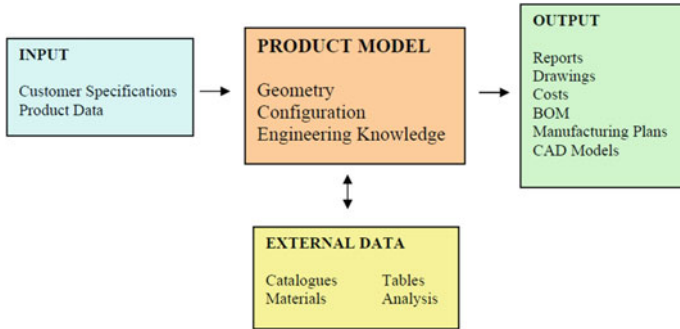


Fig. 24 The KBE system [75]

9.5 Methodology to Develop KBE Systems

The existing KBE methodologies mostly focus on systems (KBSs). For example, CommonKADS, a widely known methodology for knowledge engineering and knowledge management, is powerful but also difficult to learn and complex to use [20]. However, it was not appropriately developed for KBE applications [59]. Methodology- and tools-oriented knowledge-based engineering applications (MOKA) is another project aimed at developing a methodology to form the basis of the international standard for KBE. MOKA is based on eight KBE lifecycle steps: *identify, justify, capture, formalize, package, distribute, introduce, and use*; however, it mainly focused on the *capture* and *formalize* steps [59, 89].

Knowledge capture is the first KBE process, intended to elicit knowledge from experts or extract it from other sources (such as documentation). This collected knowledge can be structured in an informal model. Correctness and completeness are checked, and the ambiguity of language expression is eliminated. Only after such post processing will the collected knowledge become understandable and usable. There are several ways to elicit knowledge, depending on the knowledge source. The most widely used method of extracting knowledge from experts is to interview them. Another common method is to use data mining technology, which originates from the artificial intelligence domain, to extract knowledge from documents. For example, MOKA elicits knowledge from both experts and documents with its engineering domain ontology, which enables the identification of a large number of knowledge objects [89].

Knowledge formalize is the process of transforming knowledge into a neutral and formal model that can be embedded in any KBE applications. This is a process to convert knowledge into a computer-interpretable representation that facilitates encoding into a computer program. In MOKA, knowledge is represented with the MOKA modeling language (MML), which is adapted from the UML.

9.6 Implementation of KBE in Industrial Practices

Product configuration management (PCM) in product development is a good example of KBE application areas [88]. PCM provides the tools to translate the engineering specifications and validation logic for option-oriented, customer-specified product lines into a centralized PCM knowledge base. The PCM knowledge base is made up of multiple rule types, data tables, and algorithms that are maintained independently and associated with a product line, allowing PCM logic to be shared across multiple product lines as required.

Take the example of Toyota. It launched the brand Scion with two models, XA and XB, and more than 40 types of accessories for customization. Customers can refer to detailed information offered online or from dealers to customize the configuration (color, transmission, exterior, interior, wheels, and sound). Once the order is finished, the customized car will be ready for pickup. The same KBE technique has been applied to other Toyota products, such as Camry and Corolla [88].

As to commercially available implementation platforms, Siemens NX knowledge fusion [65] is a typical and fully integrated knowledge-based engineering tool that permits knowledge-based extension of NX by the end user. Compared to traditional KBE technologies, the tight integration of Knowledge Fusion into the NX digital product development system provides a significant industrial advantage. Knowledge fusion permits the creation of powerful applications that take advantage of engineering knowledge. It supports the capture and reuse of design intent and user intelligence to increase design speed and productivity, while intelligently controlling change propagation [84].

As illustrated in Fig. 25, companies can customize NX to include a set of features common to their particular design practices with user-defined features. They can be used to streamline the design process, promote reuse, and ensure that product designs follow common methods and utilize standard design components. User-defined features can take advantage of a robust set of additional knowledge fusion capabilities, which is a built-in easy-to-use design scheme that allows the designer to create rules to capture the design intent and the rationale behind design decisions. This capability is made possible by allowing rules to be attached to the user-defined feature. These rules can be used to alter the geometry, location, and even the selection of the appropriate user-defined features based on model conditions.

9.7 Future Research Issues

Knowledge technology and Artificial intelligence have a long history of development, but KBE applications specific to product engineering are still new and have yet to mature. For example, KBE systems are only used by big companies like Boeing and Toyota; tools for building KBE applications such as the methodology for

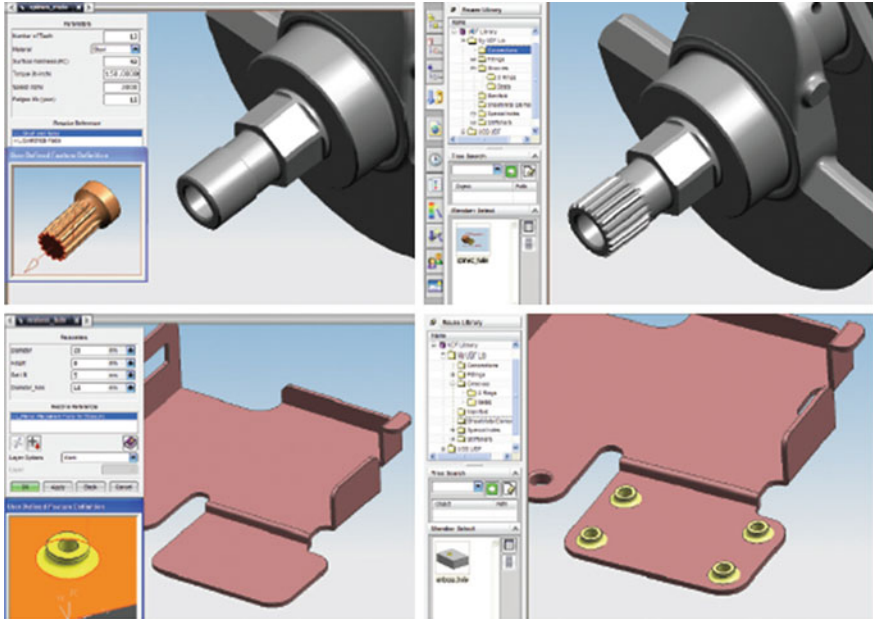


Fig. 25 User-defined features supported by knowledge fusion [84]

knowledge-based engineering applications (MOKA) [89] are similarly geared toward big companies. Little research effort has been spent on building KBE systems for small and medium-sized enterprises [59].

Transparency of reasoning procedure in KBE applications is also in need of improvement. At present, most KBE applications operate in a black box: nobody knows what the justifications are for deriving the results. If the built-in reasoning logics are faulty, the results will be very misleading. A more user-friendly, flexible, and adaptable knowledge base is needed.

10 Summary

Given the growing industrial demand for engineering information integration, there should be a systematic and scalable approach to developing a uniform implementation platform for informatics solutions that can deal with real world diversity and complexity. This chapter presented a set of challenges that require a new paradigm to address them. Among the many challenges, in-house knowledge representation and implementation, associative information-sharing and management, and cross-domain data and semantics integration (such as CAD and CAE integration) by consistent referencing and constraint management all offer new grounds for research and development.

This chapter tried to address these challenges by investigating their complex requirements and exploring some initial conceptual solutions. It seems that, theoretically, extended feature technology offers the requisite flexibility in associating entities from different domains with different levels of granularity. One of the application industries is oil and gas, where chemical process engineering is the leading field. A special section was dedicated to discussing informatics solutions in this field. Every chemical process engineering project is a complicated task, requiring the collaboration of engineers from different domains, such as chemical process engineering and mechanical engineering. Due to the complexity and close associations among the activities involved in the chemical process engineering project, interoperability is a major issue. A NDF (such as STEP) and traditional feature technology can only deal with structural heterogeneity.

To enhance interoperability, especially on the semantic level, an innovated integration framework was proposed based on semantic modeling and feature technology, and was designed to support semantics, patterns, association, and change propagation in the chemical process engineering project. This unified semantic integration framework is an open architecture and is designed to integrate a number of software tools with a common system infrastructure of consistent information referencing and updating mechanisms via a cluster of features. In that section, those associations between semantic features were illustrated by an example of equipment design, which provided a generic semantic representation of the associations across multiple design stages in the chemical process project lifecycle.

Compared to conventional design, ideally the lifecycle of the design phase can be shortened and design consistency can be more easily maintained with a significant decrease in modeling effort. In the future, constraint management and the consistency mechanism need to be enhanced, and knowledge extraction and management need further research effort.

References

1. Babic BR, Nesic N, Miljkovic Z (2008) A review of automated feature recognition with rule-based pattern recognition. *Comput in Ind* 59:321–337
2. Babic BR, Nesic N, Miljkovic Z (2011) Automatic feature recognition using artificial neural networks to integrate design and manufacturing: review of automatic feature recognition system. *Artif Intell for Eng Des, Anal Manuf* 25:289–304
3. Bayer B, Marquardt W (2003) A comparison of data models in chemical engineering. *Concurrent Eng* 12:129–138
4. Bayer B, Marquardt W (2004) Towards integrated information models for data and documents. *Comput Chem Eng* 28:1249–1266
5. Belaziz M, Bouras A, Brun JM (2000) Morphological analysis for product design. *Comput Aided Des* 32:377–388
6. Bernard A, Deglin A (2000) Knowledge-based environment for the generation of rapid product development processes. *J Manuf Sci Prod* 3:167–174

7. Bianconi F, Conti P, Diangelo L (2006) Interoperability among CAD/CAM/CAE systems: a review of current research. In: Proceedings of the conference on geometric modeling and imaging: new trends, Washington, D.C
8. Bidarra R, Bronsvoort WF (2000) Semantic feature modeling. *Comput Aided Des* 32:201–225
9. Bidarra R, Berg E, Bronsvoort WF (2001) Collaborative modelling with features. In: Proceedings of ASME design engineering technical conference, Pittsburgh
10. Bronsvoort WF, Jansen FW (1993) Feature modeling and conversion: key concepts to concurrent engineering. *Comput Ind* 21:61–86
11. Bronsvoort WF, Noort A (2004) Multiple-view feature modeling for integral product development. *Comput Aided Des* 36:929–946
12. Bylund N (2012) Modes, methods and tools for car body development. Licentiate thesis, Lulea University of Technology
13. CATIA (2012) <http://www.3ds.com/products/catia/>. Accessed 19 Aug 2012
14. CATIA/CAA (2012) <http://www.3ds.com/plm-glossary/caa-v5/>. Accessed 19 Aug 2012
15. Chapman CB, Pinfold M (2001) The application of a knowledge based engineering approach to the rapid design and analysis of an automotive structure. *Adv Eng Softw* 32:903–912
16. Chen G, Ma YS, Thimm G, Tang SH (2004a) Unified feature based integration of design and process planning. In: Hinduja S (ed) Proceedings of the international MATADOR conference. Springer, London
17. Chen G, Ma YS, Thimm G, Tang SH (2004) Unified feature modeling scheme for the integration of CAD and Cax. *Comput Aided Des Appl* 1:595–601
18. Chen G, Ma YS, Thimm G, Tang SH (2005) Knowledge-based reasoning in a unified feature modeling scheme. *Comput Aided Des Appl* 2:173–182
19. Chen G, Ma YS, Thimm G, Tang SH (2006) Associations in a unified feature modeling scheme. *ASME Trans J Comput Inf Sci Eng* 6:114–126
20. CommonKADS (2012) <http://www.commonkads.uva.nl/>. Accessed 1 Oct 2012
21. Couper JR, Penney WR, Fair JR et al (2004) Chemical process equipment: selection and design, 2nd edn. Elsevier, Burlington
22. Cutkosky MR et al (2006) PACT: an experiment in integrating concurrent engineering systems. *IEEE Comput* 26:28–37
23. Deng YM, Britton GA, Lam YC, Tor SB, Ma YS (2002) Feature-based CAD-CAE integration model for injection-moulded product design. *Int J Prod Res* 40:3737–3750
24. Fischer A, Wang KK (1997) A method for extracting and thickening a mid-surface of a 3D thin object represented in NURBS. *J Manuf Sci Eng Trans ASME* 119:706–712
25. Fonseca F, Egenhofer M, Davis C, Câmara G (2002) Semantic granularity in ontology-driven geographic information systems. *AMAI Annals Math Artif Intell* 36:121–151
26. Foucault G, Cuilliere JC, Francois C, Leon JC, Maranzana R (2008) Adaptation of CAD model topology for finite element analysis. *Comput Aided Des* 40:176–196
27. Francois V, Cuilliere JC, Gueury M (1999) Automatic meshing and remeshing in the simultaneous engineering context. *Res Eng Des* 11:55–66
28. Gabbert U, Wehner P (1993) Steps towards CAD–FEA integration. *Eng Comput* 9:17–26
29. Gao J, Zheng DT, Gindy N (2004) Extraction of machining features for CAD/CAM integration. *Int J Adv Manuf Technol* 24:573–581
30. Gao J, Zheng DT, Gindy N (2004) Mathematical representation of feature conversion for CAD/CAM system integration. *Robot Comput Integr Manuf* 20:457–467
31. Gordon S (2001) An analyst's view: STEP-enabled CAD-CAE integration. In: Proceedings of presentation materials of NASA's STEP for aerospace workshop, jet propulsion laboratory, Pasadena
32. Gujarathi GP, Ma YS (2010) Generative CAD and CAE integration using common data model. In: Proceedings of the 6th annual IEEE conference on automation science and engineering, Toronto

33. Gujarathi GP, Ma YS (2011) Parametric CAD/CAE integration using a common data model. *J Manuf Syst* 30:118–132
34. Han JH, Pratt M, Regli WC (2000) Manufacturing feature recognition from solid modes: a status report. *IEEE Trans Robot Autom* 16:782–796
35. Han S, Lee T (1999) Information sharing between process and engineering design activity in CAD environment. *Comput Chem Eng Suppl*:S573–S576
36. Hao Q, Shen WM et al (2006) Agent-based collaborative product design engineering: an industrial case study. *Comput Ind* 57:26–38
37. Hu KM, Wang B, Liu Y, Huang J, Yong JH (2012) An extended schema and its production rule-based algorithms for assembly data exchange using IGES. *Int J Adv Manuf Technol* 58:1155–1170
38. ISO 6983-1 (1982) Numerical control of machines-program format and definition of address words—part 1: data format for positioning, line motion and contouring control systems. International Organization for Standardization, Geneva
39. ISO 10303-42 (1994) Industrial automation systems and integration: product data representation and exchange—part 42: integrated generic resources: geometric and topological representation. International Organization for Standardization, Geneva
40. ISO 10303-1 (1995) Industrial automation systems and integration: product data representation and exchange—part 1: overview and fundamental principles. International Organization for Standardization, Geneva
41. ISO 10303-212 (2001) Industrial automation systems and integration: product data representation and exchange—part 212: application protocol: electrotechnical design and installation. International Organization for Standardization, Geneva
42. ISO 10303-232 (2002) Industrial automation systems and integration: product data representation and exchange—part 232: application protocol: technical data packaging core information and exchange. International Organization for Standardization, Geneva
43. ISO 14649-1 (2003) Industrial automation systems and integration: physical device control—data model for computerized numerical controllers—Part 1: overview and fundamental principles. International Organization for Standardization, Geneva
44. ISO 10303-227 (2005) Industrial automation systems and integration: product data representation and exchange—Part 227: application protocol: plant spatial configuration, 2nd edn. International Organization for Standardization, Geneva
45. ISO 10303-239 (2005) Application protocol for product life cycle support: HTML document and electronic inserts (includes access to additional content). International Organization for Standardization, Geneva
46. ISO 10303-221 (2007) Industrial automation systems and integration: product data representation and exchange—Part 221: application protocol: functional data and their schematic representation for process plants. International Organization for Standardization, Geneva
47. ISO 10303-238 (2007) Industrial automation systems and integration: product data representation and exchange—Part 238: application protocol: application interpreted model for computerized numerical controllers. International Organization for Standardization, Geneva
48. ISO 10303-210 (2011) Industrial automation systems and integration: product data representation and exchange—Part 210: application protocol: electronic assembly, interconnect, and packaging design (2nd edn). International Organization for Standardization, Geneva
49. Ji AM, Zhu K, Huang JC, Dong YP (2011) CAD/CAE integration system of mechanical parts. *Adv Mater Res* 338:272–276
50. Kao YC, Cheng HY, She CH (2006) Development of an integrate CAD/CAE/CAM system on taper-tipped thread-rolling die-plates. *J Mater Proc Technol* 177:98–103
51. Kemmerer S (1999) STEP: the grand experience. National Institute of Standards and Technology Special Publication, USA

52. Kim J, Pratt M, Iyer RG, Sriram RD (2008) Standardized data exchange of CAD models with design intent. *Comput Aided Des* 40:760–777
53. Lam SM, Wong TN (2000) Recognition of machining features: a hybrid approach. *Int J Prod Res* 38:4301–4316
54. Lander SE (1998) Issues in multi-agent design systems. *IEEE Expert Intell Syst Appl* 12:18–26
55. Lee HC, Jhee WC, Park HS (2007) Generative CAPP through projective feature recognition. *Comput Ind Eng* 53:241–246
56. Lee SH (2005) A CAD-CAE integration approach using feature-based multi-resolution and multi-abstraction modeling techniques. *Comput Aided Des* 37:941–955
57. Li SH, Shao XD, Ge XB (2010) A kind of CAD/CAE integrated modeling technology based on feature. *Adv Mater Res* 97–101:3436–3442
58. Liang J, Shah JJ, D'Souza R et al (1999) Synthesis of consolidated data schema for engineering analysis from multiple STEP application protocols. *Comput Aided Des* 31:429–447
59. Lovett PJ, Ingram A, Bancroft CN (2000) Knowledge-based engineering for SMEs: a methodology. *J Mater Process Technol* 107:384–389
60. Ma YS (2009) Towards semantic interoperability of collaborative engineering in oil production industry. *Concurrent Eng* 17:111–119
61. Ma YS, Tong T (2003) Associative feature modeling for concurrent engineering integration. *Comput Ind* 51:51–71
62. Marquardt W, Nagl M (2004) Workflow and information centered support of design processes: the IMPROVE perspective. *Comput Chem Eng* 29:65–82
63. Monedero J (2000) Parametric design: a review and some experiences. *Autom Constr* 9:369–377
64. NX (2012) http://www.plm.automation.siemens.com/en_us/products/nx/. Accessed 19 Aug 2012
65. NX knowledge fusion (2010) http://www.plm.automation.siemens.com/en_us/products/nx/. Accessed 19 Aug 2012
66. NX programming and customization (2010) http://www.plm.automation.siemens.com/zh_cn/Images/4988_tcm78-4564.pdf. Accessed 19 Aug 2012
67. Owen J, Bloor MS (1987) Neutral formats for product data exchange: the current situation. *Comput Aided Des* 19:436–442
68. Pahng GDF, Bae S, Wallace D (1998) Web-based collaborative design modeling and decision support. In: *Proceedings of the ASME design engineering technical conferences, Atlanta*
69. Perng DB, Cheng CF (1997) Resolving feature interactions in 3D part editing. *Comput Aided Des* 29:687–700
70. Pratt MJ, Anderson BD, Ranger T (2005) Towards the standardized exchange of parameterized feature-based CA models. *Comput Aided Des* 37:1251–1265
71. Pro/TOOLKIT (2012) <http://www.ptc.com/product/creo/toolkit>. Accessed 19 Aug 2012
72. Razayat M (1996) Midsurface abstraction from 3D solid models: general theory and applications. *Comput Aided Des* 28:905–915
73. Remondini L, Leon JC, Trompette P (1996) Generic data structures dedicated to integrated structural design. *Finite Elem Anal Des* 22:281–303
74. Salomons OW, van Houten FJAM, Kais HJJ (1993) Review of research in feature-based design. *J Manuf Syst* 12:113–132
75. Sandberg M (2003) Knowledge based engineering in product development. Technical report, Lulea University of Technology
76. Sequin CH (2005) CAD tools for aesthetic engineering. *Comput Aided Des* 37:737–750
77. Shah JJ (1991) Assessment of feature technology. *Comput Aided Des* 23:331–343
78. Shah JJ, Mantyla M (1995) Parametric and feature based CAD/CAM: concepts, techniques and applications. Wiley, New York

79. Shahin TMM (2008) Feature-based design: an overview. *Comput Aided Des Appl* 5:639–653
80. Shen WM, Barthes JP (1997) An experimental environment for exchanging engineering design knowledge by cognitive agents. In: Mantyla M, Finger S, Tomiyama T (eds) *Knowledge intensive CAD-2*. Chapman & Hall, London
81. Shen WM, Hao Q et al (2010) System integration and collaboration in architecture, engineering, construction, and facilities management: a review. *Adv Eng Inform* 24:196–207
82. ShuMing Gao (1998) A survey of automatic feature recognition, *Chin J Comput* 21:281–288. In Chinese
83. Siegel J (1996) *CORBA: fundamentals and programming*. Wiley, New York
84. Siemens PLM Software (2012) http://www.plm.automation.siemens.com/en_us/. Accessed 1 Oct 2012
85. Singh N et al (1997) A knowledge engineering framework for rapid design. *Comput Ind Eng* 33:345–348
86. Spatial (2012) <http://www.spatial.com/products/3d-acis-modeling>. Accessed 12 Sep 2012
87. Srinivasan V (2008) Standardizing the specification, verification, and exchange of product geometry: research, status and trends. *Comput Aided Des* 40:738–749
88. Stanford Graduate School of Business (2005) Toyota: demand chain management, global supply chain management forum. Case: GS-42
89. Stokes M (ed) (2001) *Managing engineering knowledge: MOKA—methodology for knowledge based engineering applications*. Wiley, New York
90. Sudarsan R, Fenves SJ, Sriram RD, Wang F (2005) A product information modeling framework for product lifecycle management. *Comput Aided Des* 37:1399–1411
91. Suh SH, Cho JH, Hong HD (2002) On the architecture of intelligent STEP-compliant CNC. *Int J Comput Integr Manuf* 15:168–177
92. Suh SH, Lee BE, Chung DH, Cheon SU (2003) Architecture and implementation of a shop-floor programming system for STEP-compliant CNC. *Comput Aided Des* 35:1069–1083
93. Tang SH, Ma YS, Chen G (2004) A feature-oriented framework for web-based CAX applications. *Comput Aided Des Appl* 1:117–126
94. Tseng YJ, Joshi SB (1994) Recognizing multiple interpretations of interacting machining features. *Comput Aided Des* 26:667–688
95. Ulieru M, Norrie D, Kremer R, Shen WM (2000) A multi-resolution collaborative architecture for web-centric global manufacturing. *Inf Sci* 127:3–21
96. UnifiedCAD (2012) <http://www.unifiedcad.com/>. Accessed 19 Aug 2012
97. van Dijk CGC (1995) New insights in computer-aided conceptual design. *Des Stud* 16:62–80
98. Wang H (1995) An approach to computer-aided styling. *Des Stud* 16:50–61
99. Wang XV, Xu XW (2012) DIMP: an interoperable solution for software integration and product data exchange. *Enterp Inf Syst* 6:291–314
100. Wang YD, Shen WM, Ghenniwa H (2003) Webblow: a web/agent-based multidisciplinary design optimization environment. *Comput Ind* 52:17–28
101. Wiesner A, Morbach J, Marquardt W (2011) Information integration in chemical process engineering based on semantic technologies. *Comput Chem Eng* 35:692–708
102. Wittenborn D (2004) CAD interoperability. http://www2.tech.purdue.edu/cimt/courses/cimt311/cad_interop.pdf. Accessed 13 Aug 2012
103. Xie Y, Wei J, Ma YS (2012) Multi-view feature model representation to support integration of chemical process and mechanical design. *Comput Aided Des Appl*: accepted
104. Xu XW, Wang J (2004) Development of a G-code free, STEP-compliant CNC lathe. In: *Proceedings of 2004 ASME international mechanical engineering congress and exposition, Anaheim*
105. Xu XW, Newman ST (2006) Making CNC machine tools more open, interoperable and intelligent: a review of the technologies. *Comput Ind* 57:141–152

106. Xu XW et al (2005) STEP-compliant NC research: the search for intelligent CAD/CAPP/CAM/CNC integration. *Int J Prod Res* 43:3703–3743
107. Xu XY, Wang YY (2002) Multi-model technology and its application in the integration of CAD/CAM/CAE. *J Mater Proc Technol* 129:563–567
108. Yin CG, Ma YS (2012) Parametric feature constraint modeling and mapping in product development. *Adv Eng*. doi:[10.1016/j.aei.2012.02.010](https://doi.org/10.1016/j.aei.2012.02.010)
109. Zhou X, Qiu Y, Hua G, Wang H, Ruan X (2007) A feasible approach to the integration of CAD and CAPP. *Comput Aided Des* 39:324–338
110. Zhu GuoWang (1994) An overview of feature technology on CAD/CAM, *Chin J Comput Appl* 8–12. In Chinese