Design components

White paper

Version 1.0

Submitted by:

Open Cascade S.A. (an EADS Matra Datavision company)

Contacts

Owner / Editor: **Gilles Debarbouille**

Address: Open Cascade S.A.
31, avenue de la Baltique
91954 Les Ulis CEDEX
France
Telephone: +33 1 69 82 24 88
FAX: +33 1 69 82 24 02
E-Mail: g-debarbouille@opencascade.com

Owner / Editor: **Mikhail Kazakov**

Address: Open Cascade S.A.
31, avenue de la Baltique
91954 Les Ulis CEDEX
France
Telephone: +33 1 69 82 28 79
FAX: +33 1 69 82 24 02
E-Mail: m-kazakov@opencascade.com

Executive Summary

This white paper describes the vision onto the problem of organization of engineering distributed components and declares the intention to propose another possible area of standardization within the bounds of OMG MfgDTF (Object Management Group Manufacturing Domain Task Force).

We assume, that engineering domain, currently devoted to CAD, CAM, CAE, PDM, and other areas shall have the evolution in the direction of interoperable component-oriented distributed software. The first two steps were done by the OMG MfgDTF in this direction, considering OMG CAD Services and the PDM Enablers specifications. The next step from our point of view stays in the area of multi-engineering algorithmic components, which may be considered as bricks for building multi-domain engineering system, relying on the CAD Services specification and operating in the heterogeneous distributed environment within OMG CORBA world. We presume also to support the recent OMG MDA paradigm. We believe that for the successful component-oriented software in the domain of engineering systems, the effort for standardization shall be done, in order to establish the common view on the component structure and provide required level of interoperability. We presume for such standardized engineering components to be called “Design components”.

Our intention involves two levels of “Design components”. First level – “Interconnection Service” - is abstract, domain independent, gives interconnection level for algorithms and establishes the grounds for the second level, representing domain-specific algorithmic components. Interconnection Service shall provides a set of tools and mechanisms for control and management of domain-specific algorithmic components. Domains here covers the variety of engineering needs, such as mechanical engineering, rapid prototyping, numerical analysis, machining, electrical engineering and many others. Abstract level shall provide the most common mechanisms, needed for the building of component-oriented distributed engineering system, such as data passing among components, composition of low-level functioned components into the sequence of them, producing high-level functionality and many others. Domain-oriented components shall be divided according the specific domains of use, such as mechanical or electrical engineering and provide the specification of functionality with the decomposition onto different reusable layers. Many algorithms from one engineering domain may be reused inside others and this fact requires standardization.

This papers provides the general overview of the current state of engineering software, its problems, needs and requirements according to the requests of business and current IT trends. After we try to prove the needs in Design Components and we declare benefits of Design Components within the current state of engineering software. In the next chapters, we give our considerations about the possible Interconnection Service and domain-oriented design components, focusing mainly on the mechanical engineering, which we presume to be the first area of standardization.

Table of context

Table of Figures

1 Introduction

The focus of this document is to provide preliminary information about possible development of new OMG standards in the engineering domain, dedicated to algorithmic components in different areas of engineering and production. This document outlines possible scenarios, explicates the idea of "Design Components" and defines the high level view on decomposition and reuse of engineering components, mainly for mechanical design domain.

1.1 Goal of this white paper

The goal of this document is to attract people, companies and research institutions, involved in the area of engineering and production systems, in order to discuss proposed issues, initiate "Design Components RFP (Request For Proposal)" and work out a specification of multi-engineering reusable CORBA components. The intention of authors is to build a team of industrial companies and research institution inside the OMG in order to work for such specification. This white paper does not describe any solutions, but only problematic, motivation and possible requirements for "Design Components".

1.2 Intended audience

This document was written for engineering software vendors, industrial and academic users, which require the high level of interoperability of software solutions in terms of component integration, reuse of algorithmic components among different engineering domains and optimization of current solutions by explicit decomposition of algorithms.

1.3 The Object Management Group (OMG)

The Object Management Group (OMG) is an open consortium of software vendors and users dedicated to creating and popularizing object-oriented standards for application integration, based on existing technology. The organization's charter includes the establishment of detailed service and interface specifications that provide a common framework for the development of component applications for distributed computing systems. Conformance to these specifications allows independently developed off-the-shelf software packages to participate in distributed systems designed by end-users in a heterogeneous computing environment that includes all major hardware/software platforms. OMG specifications are used worldwide for distributed applications in Manufacturing, Finance, Telecommunications, Electronic Commerce, Health Care, Public Utilities, Life Sciences Research and other domains.

OMG defines object technology as software that models the real world through "objects." These objects are the exposed representation of the attributes, relationships and methods provided by underlying software components. Object management is the design, deployment, operation and maintenance of systems based on object technology. A key benefit of an object-oriented system is its ability to expand in functionality by replacing existing components and extending the set of components (the "object instances") that provide a given interface, without altering other components of the system. Object management results in faster application deployment, easier maintenance, enormous scalability and reusable software.

Year 2000, The OMG Manufacturing Domain Task Force, one of the standards development bodies of the OMG, released a Request for Proposal for a standard set of object interfaces for basic CAD/CAM/CAE interoperability, called CAD services. July 2001 the Finalization Task Force was chartered to finalize the specification, which shall be ready until the end of year 2001.

2 Current situation of engineering software

Currently the engineering systems (CAD/CAM/CAE/DMU/etc.) are build as huge and not widely and dynamically reusable software entities. That leads to spending much time for creation, change management, configuration and other steps of software management process. Creation of multi-engineering software in the concurrent environment also requires many forces due to the absence of ready-to-use loosely coupled components, which can be quickly integrated to achieve the needed level of functionality.

Mainly this situation exist due to three facts:

- There are very few adopted and working standards for dynamic runtime interfaces in engineering
- There is no common declared view onto the functionality, used in the multi-engineering solutions, which may be provided in a standard way by vendors.
- Standardization of engineering domain is very hard due to the extremely complex behavior and data structures of such software.

A decade ago, the DMAC specification was started by the set of companies in order to define DCOM architecture for CAD software, but this effort was stopped after some years of development without any final specifications. At the same time, other attempts to build a small consortium were done, but the result was the same.

Recently, in the bounds of the OMG MfgDTF, two standards are in adoption process: PDM enablers V2.0 specification, which attempt to standardize the area of product data management and CAD Services V1.0 specification, which defines the basic interoperability level for CAD/CAM/CAE interoperability. Both standards define CORBA-based interfaces for platform-independent, interoperable and distributed use of declared functionality.

However, these standards are not enough to define the engineering system, built on top of loosely coupled distributed components. New standards and specifications of domain-dependent problems are required. From the authors' point of view, these standards shall continue the common direction of mentioned above specifications and declare platform-independent IDL interfaces and UML descriptions. This will allow using of specifications on a variety of platforms and technological bases, as declared recently in the MDA paradigm of the OMG.

3 CAD Services specification

Until the end of year 2001, the OMG CAD Services V.1.0 standard shall be ready. CAD Services submission proposes an interface standard for Mechanical Computer Aided Design (CAD) systems that enable the interoperability of CAD, Computer Aided Manufacturing (CAM) and Computer Aided Engineering (CAE) tools. The aim is to provide users of design and engineering systems the ability to seamlessly integrate, best-in-class, software across a wide variety of CAD/CAM and CAE applications through CORBA interfaces. These standard interfaces enable a distributed product design environment that includes a variety of CAD systems.

CAD Services V 1.0 proposal focuses on establishing Mechanical CAD system interfaces that provide Geometry and Topology data to Analysis and Manufacturing applications and tools. The intent is to establish a series of basic high-level engineering interfaces that do not require low-level data structures to answer mechanical engineering queries. To avoid many of the problems associated with data translation, this proposal provides CORBA interfaces with consistent functionality across native CAD implementations.

From our point of view, CAD Services specification defines the strong establishment for future definition of other standards, which will involve high-level functionality from different

engineering domains (mechanical design, numerical simulation, rapid prototyping, optic, electromechanical and many others).

CAD Services V 1.0 specification defines the basic notion of «feature», which is essential for many modern 3D modeling engines. However, specification does not speak about the way to call CAD algorithms. The possibility of dynamic association of algorithmic component to some feature looks very attractive, but for that, some effort of standardization of algorithmic components is required. Now we introduce the notion of «Design components», which represents the set of interfaces, supporting algorithmic functionality of CAx systems, allowing reuse of existing solutions and rapid building of applications and composite algorithms from this predefined set.

4 Why “Design Components”?

Thus, we may speak about the area of distributed loosely coupled algorithmic software components, working in engineering software, which we will call “Design components”. We presume that “Design components” shall be described as IDL interfaces and pretend to be standardized in the bounds of the OMG. We propose to deliver “Design Components Request For Proposal” and to establish «Joint Design Components submission team», which will start the adoption process of set specifications. Main ideas, what these specification may present are represented below. Here we consider mainly the design components, dedicated to 3D modeling domain. We assume that this domain shall be standardized before other components (See interconnection of modeling algorithms with other domains in the «Domains» chapter of this white paper). Other domains are presumed to be standardized later and mentioned in this white paper either to show interconnection with 3D modeling or for generalization purpose.

CAD/CAM systems and libraries often propose high level functionality and/or very low level functions (See Figure 2). High level functionality is quite practical to use the global functionality of the system. Generally, what a user can do in the interactive mode is possible in a batch mode using these APIs.

The low-level functions are used to program applications, which directly manipulate basic entities like geometric and or topological entities. The main area is the programming of data exchange interfaces, which read standards files like IGES, STEP or other formatted files. If now the purpose is to develop a new specific application using an existing kernel the two previously described levels do not allow doing it in an optimized way.

The high level just allows the use of the existing functions. If the user wants to create a new functionality, he can not to do it because he has no access to the basic functionality to manipulate correctly the data and call the elementary computational functions.

Here we can give the notion of intermediate level functions, which represent the decomposition of high level functions onto their composite parts. The low level generally allow the user to do what he wants but the risk is to develop functions which already exists in the kernel. It is really a lost of time and money. It is the reason why the intermediate level, is useful for developing application in an optimized way. Here we consider, that the performance aspect is very important in the field of modeling

As it was mentioned above, engineering software now does not provide common and standard reusable solutions, consisting from the loosely coupled components. However, such solution is required to decrease the time and cost of engineering software and to achieve the higher level of market satisfaction in appropriate time. There are great examples of telecommunication, transport regulation and other domains, where loosely coupled CORBA components lead to such results. We strongly believe that the same reusable solutions shall be implemented in the engineering domain. For that, the set of worldwide-adopted specifications shall be created in addition to existing CAD Services, PDM Enables, STEP and other specifications. We propose Design Components for such standardization, since algorithms

represent the biggest part of engineering software (especially in CAD and CAE areas). It's necessary to mention, that many software vendors tried to apply CORBA, DCOM and other technologies to their products, but lake of standardization and reuse of components of other vendors lead to the huge development time and future incompatibility of proposed interfaces.

Standardization aspect shall provide a strong motivation for software vendors to be compatible and reusable in their standards. Applying of component-oriented paradigm and description in terms of IDL interfaces will guaranty loosely coupling, distribution, platform independence and will be well integrated in the future OMG MDA infrastructure and many software architectures and techniques, supported by OMG, such as CORBA, Java Beans, DCOM, etc.

In this paper, we discuss some aspects of possible definition of Design Components and we propose the open discussion among software vendors, research institutions and industrial users, which will be expressed in the future OMG MfgDTF Design Components RFP (Request For Proposal).

5 General Considerations

5.1 Possible architecture

Since Design Components are desired to work in platform independent distributed environment, we consider standard conception, such as CORBA, JavaBeans or similar as context of the work. CORBA paradigm gives also language interoperability and since mostly engineering systems are implemented in C++, we consider this platform as a basis for the description. Certainly, during the RFP preparation, the recent OMG MDA approach shall be taken into account. Relying on existence of CORBA-based CAD Services and PDM Enablers specifications, we presume for Design Components the compliance also with existing CORBA Services and CORBA Facilities. Figure 1 provides us with the definition of possible Design Components specifications in this context:

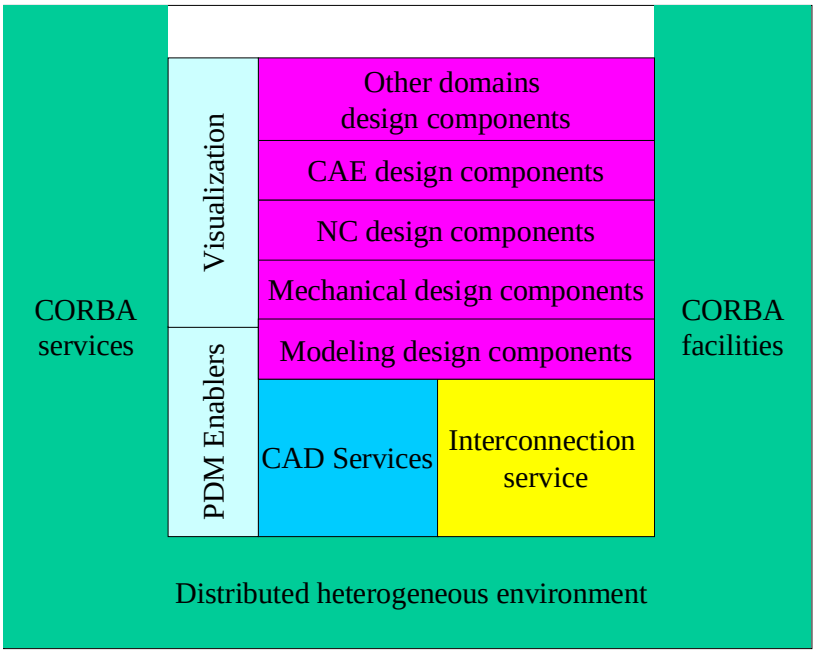

Figure 1. Possible Design Components specifications in the context of CORBA platform

Here we distinguish the layer of Interconnection Service, which is in charge to provide generic domain-independent basis for specific algorithmic components. Interconnection

Service proposes the functionality of dynamic composition of algorithms, registration of new algorithmic components, runtime extension of the software by new components and their loading and gives the abstract base for deriving new platform-dependent components.

After we presume, that each domain of engineering may be expressed in a specification, related to the domain. Domains may be:

- 3D modeling
- Numerical analysis
- Optical domain
- Electrical domain
- Machining
- Rapid prototyping
- Reverse Engineering
- and many others.

We presume, that both Interconnection Service and domain-dependent Design Components rely on the CAD Services specification, which gives the necessary level of topology-geometry interoperability and provides general structure and extension mechanisms for CAD models management. PDM enablers specification is given on this picture only for completeness. The compatibility with PDM enablers is extremely necessary on CAD Services level in order to provide document and model management. We do not see any compliance points among Design Components and PDM systems. We can mention another possible specification, which define Visualization functionality of CAD/CAM/CAE data. This specification will be useful for completeness of interoperable distributed solution in engineering area and shall be discussed for possible standardization. As for PDM Enablers, this specification has no direct relations with Design Components, but would depend on CAD Services.

From the domain component we suppose, that 3D modeling Design Components shall be worked out and standardized at the first stage. Necessity of these components is bound to the wideness of use in engineering area and high probability of reuse of 3D modeling algorithms, such as sub-shapes intersection in different engineering domains.

However, other domains shall be also considered for possible standardization or definition of flexible extension mechanism, allowing each vendor to be easily integrated inside third party software and allowing high level of interoperability. Surely in case of extension mechanism, it is necessary to rely on some standards for engineering data.

5.2 Interconnection Service

As it was mentioned above, we presume Interconnection Service as a basic layer for all possible engineering component specifications. This layer shall define the basic architecture and propose guideline for creation of algorithmic components, working with complex data (we presume that engineering data is very complex and highly structured for example the structure of topological shape). Interconnection Service shall define a mechanism of description of «big» high level algorithms by the sequence of more low level algorithms and mechanisms for assembly of new «composite» algorithms from basic ones. Here two examples may be done from 3D modeling domain. Let us consider Boolean Operations algorithms from 3D modeling domain. Figure 2 describes the decomposition of Boolean Operations into intermediate level components. Client may use either entire Boolean Operation implementation, or to use three steps of intermediate level of algorithms. Each Boolean Operation (Union, Difference, and Intersection) differs from other only by the last step, which takes approximately 10% of computation time. Here the knowledge about decomposition may give a big performance improvement in parametric modeling, where for instance, the type of Boolean operation may be changed without recomputation of other two

parts (90 % of computation time). Thus we assume having a mechanism in Interconnection Service, which will be able to provide the freedom for the end user either to call the complete implementation of the boolean operation or to use a sequence of intermediate-level components.

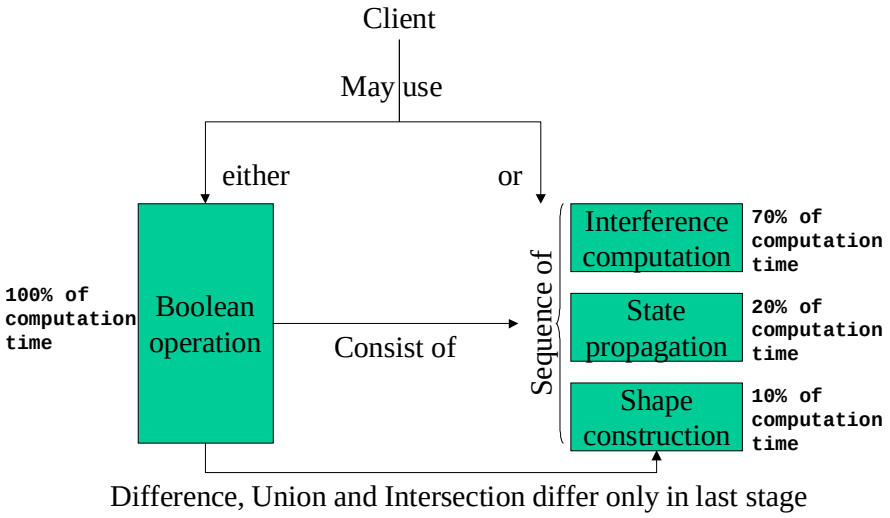

Figure 2. Example of decomposition of Boolean Operations

Second example is a composite algorithm, which is creation screwbolt hole, which requires to remove cylindrical part of material from some body and after remove conical parts from the top and bottom faces of cylindrical parts. Such algorithm (requires one cylinder removal and two cones removal) is widely used in the parametric feature-based modeling and may be easily composed «on fly» by sequential combination of two well-known components. For that, Interconnection Service has to provide the unique way to pass parameters from one algorithmic component to another during dynamic composition of components without additional coding.

Consequently, we propose to define in the Interconnection Service the possible guidelines and mechanisms for such composition and decomposition and to provide the mechanisms of data interchange among composite parts of components.

One of the most important part of Interconnection Service is the mechanism for dynamic extension, when new algorithms may be added and used «on fly» without reconfiguration and recompilation of the software. There are many specific engineering areas, where it is quite impossible to define the standard for component interfaces due to their specificity or difference among vendors. Here, the common mechanism to connect different components, working in standard data is required.

5.3 Domain oriented design components

The open discussion shall be started, which engineering areas and algorithms interfaces shall be standardized. We propose to start the standardization from 3D modeling domain. Almost all the vendors provide quite similar functionality in this area and there is a big chance to achieve interoperability of 3D modeling solutions by standardizing it. End user will receive

the chance to replace 3D modeling components from different vendors depending of conditions he needs (cost, Quality of Service, etc.).

In the 3D modeling domain we propose to specify several layers of algorithms and to provide their interconnection (see Figure 2 for example of BO). The general idea is to split the access to the data and the services or the algorithms by defining separated Application Programming Interface (API). The main interest is, to be completely independent of the implementation of the data structure and in addition gives the possibility to preserve partial results inside the algorithms.

Sometimes it is quite difficult to decide in which object an algorithmic method must be attached essentially due to the diversity of the types of argument. When an algorithm returns multiple solutions, it is better to have a separate class to describe the algorithm.

Having the possibility to query an algorithm is more powerful and never destroys the objects passed in argument. In these condition it is better to implement separate classes from data structure classes and in consequence separate Application Programming Interface (API).

Design Components could be considered as elementary blocks, which are assembled to make an application. These blocks are either algorithmic parts or parts to manipulate Data Structure. To be able to optimize an application it is necessary to consider these two aspects. A global algorithm cannot be optimized if it is implemented independently from the data it manipulates. Sometime data must be put in a special format to be able to access them faster. In addition, for some comparison algorithms it could be fine to use additional data to speed up the process. The last important point is the behavior of the functionality. It is essential to describe in detail the input arguments, the range of validity of the function and the possible computed results.

Thus it is necessary to identify the high level algorithms, used for 3D modeling, which are quite common for the most of 3D modeling systems and libraries. Intermediate and low level algorithms shall be defined as well. We believe that low level algorithms shall stay only on the level of semantic specification, but IDL interfaces shall not be obligatory to use in order to satisfy a performance issue. Figure 3. gives sample division of such algorithms

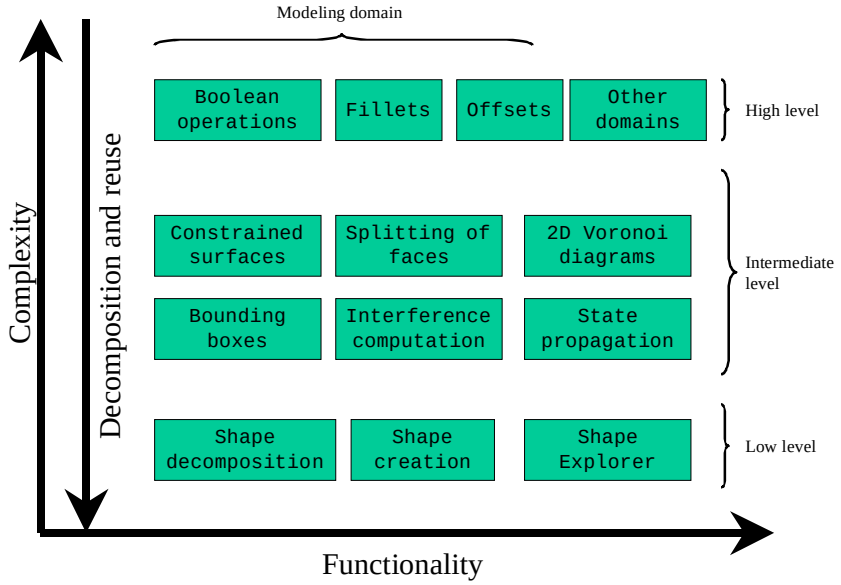

Figure 3. Sample division of algorithms by layers

After that, the guideline for composite algorithms creation shall be defined (see chapter «Interconnection Service» for example) in order to help vendors and users building the parametric feature-based software (or using other paradigm, such as CSG (Constructive Sold Geometry) or Behavioral features) with a given standard high level algorithms.

Design Components shall very well correlated with CAD Services definition of feature, which will give the interconnection and reuse of specifications inside MfgDTF.

Other domain standardization shall be discussed and authors of this white paper propose such a discussion.

5.4 Benefits for users of «Design Components»

Having a possibility of distribution, the end user may perform algorithmic parts on computers that are more powerful and servers. Client part dedicated to visualization and user interface may be located at the end-user station, when data structure may be located on a data server (CAD Services models). Data server may exchange the information with computation server, running Design Components, by performant channels giving a great enhancement of performance, sharing data and computational resources among users.

Strong definition of algorithmic interfaces in Design Components will provide the reuse of common algorithms in different areas of engineering and reuse of cost-off-the-shell algorithmic components by different vendors. Such possibility will allow rapid construction of user-specific and generic applications with reusable and easily replaceable and reconfigurable parts. With component approach and dynamic connection of algorithms, based on CAD Services specification vendors and service companies will be able to provide better satisfaction of customers and to open new markets, which are collapsed currently due to the huge cost of production of software.

Interconnection Service with the extension mechanism will be able to connect more easily some algorithms, which can't be standardized due to their specificity, and also new algorithms, which my appear on the market. Interconnection Service will play the role of a big "service level" user of CAD Services data interoperability specification by addition of "behavior interoperability".

Design Components may be stated as a big step for new solutions in the domain of enterprise concurrent engineering, which requires extremely high level of interoperability and reuse and looks for component oriented software in a highly distributed environments.

6 Domains

To show possible use cases of Design Components it seems essential to have a look on user applications in various domains. One of the first domains to take into account is the machining domain where the intelligence of the application is based on the knowledge of the operator. Some other domains are domains which request a lot of computation of intersections between curves (generally straight lines) and shells composed by trim surfaces. These domains are essentially optical domain and geophysical domain. The modeling domain itself is quite interesting due to the number of various methods used to compute hard problems like Boolean Operations or Hidden Lines Removal. Rapid prototyping domain requires specific components as well as Finite Element computation and measurement.

Here we describe some scenarios and possible decomposition and reuse of components for different domains of engineering.

6.1 3D Modeling

This domain is the most critical domain, which needs components. This domain includes a lot of comparison algorithms and basic geometrical and topological functionality. It would be nice to be able to extend the primitive generation some basic tools like the creation

of grids of patches are requested. Basic tools like classification of points in regard of a face or a solid are useful. An interesting tool is the exploration of a solid or a shape by adjacent components.

6.2 Design

This domain based on 3D Modeling domain integrates technological data. The goal is to design industrial products integrating multi technological aspects like mechanical, electrical, software and so on.
The main components that could be defined are components to create and fill the design data model, the components to check the coherence for a technological domain or the global coherence.

6.3 Machining

For the tool path generation the following items are requested:

- Hatching which could be the support of the tool path
- Offsets which could be an other type of support for the tool path and necessary to do contouring
- 2D Voronoï diagram is a powerful algorithm which remove the de looping of offsets. In addition, when the user wants multiple offsets it is a good way for global optimization of the algorithm.
- Trim of contours is necessary to complete the tool path by linking of hatches.
- Salesman problem is a useful algorithm to link elementary tool paths in the optimal way. For drilling when there is hundred of holes it is necessary to obtain the shortest tool path.
- Computation of remaining parts to machine is useful to compute the following sequence of machining. In this case Boolean operation are in the worst condition essentially due to coincidence between edges or faces of the rough part and the volume of material to remove.

6.4 Optic, geophysics and radar signature

The main problem of geometrical or topological algorithm is the hardness of the algorithm. To reduce it, a good solution is to reduce the number of computation by finding a law, which assume that there is no solution. The first solution is to compute the interference between the bounding boxes. If there is no interference, it is not useful to do the complete computation.

Another solution is to localize the elements inside voxels and after for a given voxel compare the elements inside this location. Often the same computation is done several times inside a global functionality. It is generally the case when an algorithm has to compute intermediate data to speed-up the process. If this part is accessible independently from the general algorithm, it is then possible to remove over computation. In some case of multiple computation, the current computation could be done by extrapolating the result of the previous one. Generally, the process is grandly speeded-up.

6.5 Rapid Prototyping

Rapid prototyping generally uses slices on tessellated models. For each slice sweep algorithm like hatches are used. The tessellation component is requested to output an exact 3D model in the right format. To provide slices a component to compute planar sections is requested Offsets or hatching components are useful to fill each slices.

6.6 Numerical analysis

Numerical analysis domain requires presence of other basic data type, which is mesh. Numerical analysis domain may use needed components from other areas (intersections, shapes decomposition, etc.) in any case, but the definition of Numerical Analysis components is currently restricted, since there is no official standard for meshes. In the future, the evaluation shall be done in order to create the standard IDL specification to access meshes and provide domain specific physical parameters on mesh elements (something like CAE Services).

6.7 Measurement

Measurement machines pick coordinates of control points on a part and rebuild the geometrical model. To do that a Surface Reconstruction component is requested. If elementary surfaces are built, first a sewing component could be useful to compute adjacent faces and to remove gaps along the edges.

7 Examples of 3D Modeling components

7.1 Preliminary evaluation

Applying given conception to the domain of modeling components, we can presume the next list, which gives some distinguishing of algorithms, widely used in 3D modeling, by levels. This is a preliminary evaluation, mentioned here for an example.

Low level components

The definition of low-level components is necessary for completeness of description, but from our point of view, providing of IDL interfaces for components on this level may require many distributed transactions and may be considered as optional.

- Bounding boxes
- Voxels
- Adjacent faces explorer
- Topological explorers
- Algorithms for surfaces
- Surface reconstruction by points
- Elementary intersections
- …

Intermediate levels components

- 2D Voronoï diagram
- Silhouette lines
- Splitting of faces
- Constrained surfaces
- Bounding boxes
- Shapes interference
- …

High level components

- Boolean operations
- Advanced intersections
- Fillets
- Offsets
- Sweep
- Hidden lines Removal
- …

7.2 Composite components

We can preliminary identify the list of components, which can be considered as composite, i.e. defines some specific functionality, which may be easily expressed in terms of components of more lower level.

- Screwbolt hole
- Assembly checker
- …

8 Preliminary recommendations for RFP

Here we try to state some recommendations for possible RFP, which are helpful from authors' point of view

It is necessary to define the types of data, which will be used for process of interaction among user and design components and components themselves. Preliminary, the BRep data structure is required. This data structure has already been standardizing in the bounds of CAD Services specification. From our point of view, Mesh generic data structure also shall be standardized. Mesh data structure and interoperability may be defined in a separated specification (something like CAE Services). Other types of complex engineering data to standardize inside Design Components shall be defined in the RFP.

In the RFP it is necessary to state the domains, which shall be evaluated for standardization and to define generic functionality, that shall be covered by Interconnection Service level. Design Components shall very well correlated with CAD Services definition of “feature”, which will give the interconnection and reuse of specifications inside MfgDTF.