Towards Ubiquitous Data Access Service of 3D Part Library in Cloud Scheme with Host Independence

Wei Liu (liu-wei@sjtu.edu.cn)
Shanghai Jiaotong University
Chuipin Kong
Shanghai Jiaotong University
Qiang Niu
Shanghai Jiaotong University
Xionghui Zhou

Original Research

Keywords: part library, cloud design, cross-platform, 3D library, host independence

Posted Date: February 12th, 2021

DOI: https://doi.org/10.21203/rs.3.rs-189810/v1

Version of Record: A version of this preprint was published at The International Journal of Advanced Manufacturing Technology on April 2nd, 2022. See the published version at https://doi.org/10.1007/s00170-022-09138-8.

Abstract

Most current part libraries are created and deployed in specific usage environments or CAD platforms, which inevitably brings obstacles in the share and exchange for part information. To reduce repeated development and provide a uniform interface for designers in different sites, a 3D part library in cloud scheme is presented in this paper. Host programs with predetermined specification can access the part data through an adapter according to their customized requests with uniform interfaces, which constructs a ubiquitous service. To realize host independence, the part models are created in a native ACIS modeler, and then they are converted into 3D files in various formats for practical needs, finally these files are imported into CAD systems or other platforms in real designs. The whole framework can be divided into three components, namely, PLS(Part Library Service) provider, PLS adapters and hosts. PLS provider is the kernel of 3D data access service in cloud scheme, while PLS adapters serve as the bridges that connect PLS provider and hosts, and the PLS can be grafted on various applications including current mainstream CAD systems as a plug-in module or run on the websites or even mobile terminals. The PLS provider is deployed and maintained on cloud and users can acquire remote part information within a local ongoing project. In the detailed construction of this part library, diversiform knowledge for part parameters and structures is implanted to define the geometry and rule constraints in the 3D modeling, with which the backstage has the ability of conveniently editing the information in the part library for better upgrade and contrapuntally services. The concept has been implemented within a PaaS framework to provide the ubiquitous 3D part data access, which has been successfully applied in a large number of manufacturing enterprises, and accumulates considerable practical cases.

1. Introduction

Recent years have seen a great advancement in product design with various information platforms. The increasingly richer design patterns and terminals make it possible for users to carry out researches ubiquitously, which also puts forward higher requirements on the related engineering software. As an auxiliary module, part library plays an important role in structure design by providing the predefined parts, which helps improve its efficiency. It is therefore a good idea to deploy a general part library for widespread designers and facilitate their work, for the fact that with the global product customization and specialization, 3D models of parts or products in limited number and series can hardly completely satisfy various customers' increasing requirements.

To expediently and agilely access to the parts information, several part libraries or similar systems have been presented in recent years, some of which are developed by part suppliers to promote their goods. Meanwhile, some manufacturing enterprises also establish independent modules or subsystems in PDM/ERP for an effective management of the parts from different suppliers[1−5].

From the view of implementation, developers of part libraries often construct their applications on a certain CAD platform to effectually utilize the built-in APIs or existing functions. Thus, much work can be saved during the constructing periods, while on the other hand, the libraries inevitably depend heavily on the exact host CAD platform and cannot be conveniently transplanted to another. Therefore, these part libraries are hard to be extended, transplanted, exchanged or shared by other systems in the stage of upstream and downstream partners. As a result, the individual part library is enclosed to form an isolated island and doesn't hold the ability of communicating with each other. Thus, information redundancy, mutual contradiction or repeated developments are the frequent troubles for the designers or enterprises. In addition, they rarely support the idea and mechanism of providing their independent functions alone, that is, without the supports from any CAD platform. An independent framework and lightweight deployment package of part library will great help run on the website and mobile terminal.

Cloud computing provides an excellent means for the solution of this issue[6]. IaaS, PaaS and SaaS are three categories of successively progressive services on cloud. With the cloud technologies, many applications can be deployed on the remote machines in a one-off means and provide ubiquitous data access services for the dispersed designers synchronously.

With the scheme of cloud, the basic functions of part library are implemented in the server, and on each client, an adapter also need be deployed to access local applications with function customization. The adapter, on which the services of cloud rely on, is often developed and seamlessly integrated in the clients while receiving and exchanging part information from cloud.

The services provided by the backstage program in cloud scheme is to create CAD part models in real-time according to the inputs of the remote users. The term “service” in the context means that it is open to any application in pre-defined rules. The service bridges the gaps among users, 3D platforms or other host systems, with which the clients can expediently invoke the interfaces for their own aims. For various hosts, especially for heterogeneous CAD software, the file formats are not open yet, which gives rise to an urgent problem that the native models are nearly impossible to be created and provided without the exact software. As an alternative solution of this issue, a cross-platform (or namely, CAD host-independent) framework for 3D part library is deeply investigated. The 3D part models are firstly created with ACIS in our implementation, and then InterOP is adopted to transfer them into various formats, such as stp, igs, x_t, hsf, etc, for designs on different CAD platforms, realizing “Created once, used almost everywhere”, which accords with the idea of cloud, especially of PaaS.

In addition, to effectively manage the parameters in an exact definition of 3D models, the methods to implant design knowledge into the parts is also investigated, with which the server can easily edit or customize the library according to practical needs. Flexible and effective representation of diversiform knowledge is an important issue in the reuse of existing geometrical, structural, or logical information in the stage of model edit, which is also emphasized in our work.

The deployment of part library in PaaS scheme is based on the cloud environment, as all functions should be accessed with an adapter including the website. It is a good attempt for this field in industry information. To our knowledge, there is not such a part library devoted to participat in the real design from remote server instead of only providing model download. The rest of this paper is organized as following. In next section, we will review the latest advancements in this field. The architecture and methodology will be presented in Sect. 3 and the detailed technologies in its implementation will be deeply investigated in Sect. 4. In Sect. 5, the services construction provided in cloud scheme will be described and several clients that communicate with the PaaS server will be introduced in Sect. 6. Finally, the conclusion and further plan will be given in Sect. 7.

2. Related Work

The rapid advancements in the technologies of cloud computing have been profoundly reshaping the characteristic of modern design and manufacture in that dynamically configurable and virtualized resources are provided as a service over the Internet. The traditional desktop mode can no longer adapt the new situation of distributed and rapid design pattern, and the users will seek and obtain their needs from all over the world. Cloud platform implements the feasibility and basic infrastructures for data and service sharing from the remote providers. Part libraries, which have been deeply investigated and widely applied in the structural design as a primary foundation, are an excellent cut-in field. A uniform part library has the ability of supporting multiple different designing scenes synchronously. As there are few reports about part library yet based on cloud now, the section will carry on the review on the related fields.

2.1 From Cloud Computing to Cloud Manufacturing

Cloud computing ushers in its prosperity in recent years and has taken deeply impact on software including industrial environments. It is emerging as one of the major enablers for design/manufacture, transforming traditional design/manufacturing pattern, helping align product innovation with business strategy, and creating intelligent factory networks that encourage effective collaboration.

The philosophy of "Design Anywhere, Manufacture Anywhere (DAMA)" has emerged in recent years[7]. Cloud computing provides a critical support in the implementation of DAMA. Internet-based or networked manufacturing, which is frequent confused with cloud manufacturing, mainly refers to integration of distributed resources for undertaking a single manufacturing task[8]. What is lacking in this type of manufacturing regime are the centralized operation management of the services, choice of different operation patterns and embedded access of manufacturing equipment and resources. Without these consideration, a seamless, stable and high-quality transaction of manufacturing resource services can hardly be guaranteed. Inspired by cloud computing, cloud manufacturing offers an attractive and richer solution, altering from production-oriented manufacturing to service-oriented manufacturing.

Cloud manufacturing can be defined as "a model for enabling ubiquitous, convenient, on-demand network access to a shared pool of configurable manufacturing resources, reflecting both the concept of "Integration of Distributed Resources (IoDR)" and the concept of "Distribution of Integrated Resources (DoIR)". IoDR is on the backstage of cloud manufacturing platform, as to identify various manufacturing resources, integrate them, and package them into uniform services of cloud computing. Thus, the categories and physical distributions are transparent and exhibit as a whole as for the clients. While the circumstances are contrary for DoIR, that is to say, a uniform interface for service provider can be tailored by the distributed clients through network with their respective purposes. The structure of the two concepts is illustrated in Fig. 1.

Figure 1 The Structure of the Two Concepts

The concept of cloud manufacturing involves nearly all aspects of the manufacturing industry, and in this section, we emphasize the review of the related work on the integrated cloud platform for CAD design software and applications.

The resources in the context of cloud manufacturing including physical categories and functional categories (including software), and after IoDR and package, all backstage resources are transferred into service providers for related clients or hosts in the cloud scheme. On the idea that applications in an information intensive manufacturing environment can be organized in a service oriented manner, Brecher[9–10] developed a module-based and configurable platform for interoperable CAD-CAM-CNC planning to combat the problems of software in homogeneity along CAD-CAM-CNC chain. Velde[11] reported a plug-and-play framework for the construction of modular simulation software to achieve a run-time configuration integration environment for engineering simulations. The embedded modules are detected, loaded and used at run-time in such architecture, the framework without needing of prior knowledge of the type and availability of components, thus providing true plug-and-play capabilities. Nessehi[12] proposed a framework to comeback the incompatibility problem among CAx that provides individual interfaces for different CAD/CAM/CNC systems much like IaaS. Mobile agent technology is also employed to support the intercommunication bus and CAx interfaces. In this system, different components of CAD/CAM/CNC chain can exchange information with another regardless of the native standards. Mokhtar[13] studied a similar manufacturing platform, using the axiomatic design theory to realize interoperability among the CAx chain and to generate a systematic roadmap for an optimum combination of data exchange via direct or indirect solution in the CAx environment. This approach provides some insight into how a design and manufacturing resource may be encapsulated and how it can be developed for cloud manufacturing. The researchers in Post-ECH[14], Korea proposed a concept of design and manufacture via ubiquitous computing technology. To support the concept, a ubiquitous product lifecycle support system is presented as well[15]. Module and agent concepts are mentioned in function of the request-find-provide chain. A unified data model, which is compliant with international standards, is utilized for data exchanges. The model represents the input and output in formation used in the lifecycle activities, in the stages of beginning, middle and end. Wang[16] proposed a distributed interoperable manufacturing platform as an integrative environment among CAD/CAM/CNC systems in a module-based structure. In order to integrate the software suites based on the requests and tasks from a user, service-oriented architecture is used. The user's requests are collected and organized as a serial of software services. From the service point of view, heterogeneous software tools are integrated as "Virtual Service Combinations" and provided to the user. In this way, software suites are embedded into operational processes.

2.2 Part Library for Service

As a primary foundation in mechanical design, part libraries have been deeply investigated in the past several years. Part coding, PLIB and GT-group technology are the main implementation. Recently, with the rapid advancement in CAD and information science, the field of part libraries exhibits newer and more powerful abilities. Parameterized 3D driving, ontology description, 3D model retrieval and distributed architecture have been widely applied in this field. Han[17] studied the parametric modeling technology in SolidWorks, and developed a standard part library for blanking die. Zhang[18] described the parametric feature and constraint driven method of establishing graphics library technology based on SolidWorks environment, and then developed 3D die standard parts database system based on three layer C/S mode. Wang[19] studied the features and feature models, emphasizing the 3D model of STEP protocol AP214, and analyzed its abstract definitions, entity relations and feature recognition in detail. The characteristics in part and library information model are also analyzed in detail. Lu[20] proposed a data dictionary based part information network representation and sharing technology. The parametric design technology of network data driven by the precise geometric expression models of the parts are given, together with the part feature model and visualization solution based on Web. From the view of system integration, Xu[21] proposed a distributed enterprise database system and a dynamic integration method in the Internet environment. A method of service encapsulation and interface construction for parts library system is thus proposed, on which a part library architecture based on web service is built. Zhou[22] developed a parts library system in a shared and private mode, in which the system structure, parts description and data model in library system were studied and set up, and based on the concept, a concrete realization is finally introduced. Cho[23–24] proposed meta-concepts with which the ontology developers describe the domain concepts of part libraries. The meta-concepts have explicit ontological semantics, so that they help to identify domain concepts consistently and structure them systematically. Jin[25] proposed a design reuse approach based on an engineering semantic web and implemented the system which does not rebuild the current parts library. A mapping relation is constructed to introduce engineering semantic information into the existing parts library. In addition, ant colony optimization is employed for the retrieval of design information, based on which a design reuse prototype system is implemented for solving the problem of design reuse.

After several years of rapid development, a large number of well-known parts libraries have been established and put into use to the public: 3DContentCentral[26], CADClick[27], Inpart[28], TraceParts[29], Web2CAD[30], etc. They supports both 2D or 3D files in various formats such as .SAT, .STP, and .iges for thousands of parts.

2.3 Deployment of Manufacturing Resource on Cloud

As a simplified version of cloud manufacturing, the deployment of manufacturing data resources on cloud is more feasible and easier to be implemented. Manufacturing data resources is a subset of manufacturing resources, including design documents, specifications, standard part library, etc.

The service system of cloud manufacturing general includes five layers, namely as, resource layer, middleware layer, core service layer, portal layer and application layer. The resource layer covers such elements as: design resources, simulation resources, production resources, test resources, integration resources, capability resources and management resources. They are abstracted upwards as two forms as virtualized manufacturing resources and service capability resources respectively. The middleware layer supports virtualization, service, access, perception and resources coordination. Core service layer, based on the interface of middleware layer, provides all important functions of cloud manufacturing, including service deployment/registration, search/matching, composition/scheduling, operation/fault tolerance, monitoring/evaluation and pricing/billing. Portal layer is a unified branch for service providers, platform operators and service users to handle the manufacturing resources and capabilities. Application layer has the ability of providing four modes: single-agent stage, multi-agent collaborative stage, multi-agent collaborative cross-stage and multi-agent manufacturing capability on demand.

The deployments of manufacturing resources including 3D part library do not merely refer to the cloud storage of the various files and documents, as the services about manufacturing data from cloud computing scheme is also an important aspect in the area. Within many enterprises, efficient query and rational access of design or manufacturing data, which may have been accumulated for a long time, is also an excellent way to enhance the efficiency.

Several famous companies such as Oracle, Amazon, Tencent and Alibaba, etc, have provided the basic infrastructure of cloud manufacturing, which brings great conveniences on their deployments. In this paper, we emphasize resource construction and service provider of part library on cloud scheme with a relatively mature and reusable computing framework.

3. Architecture & Method

Part library supports the ability of providing various different clients with their needed parts in 2D or 3D formats with corresponding specifications. A prerequisite of this scheme is that the 2D or 3D CAD files can be effectually created according to the practical needs, which can be categorized into two groups: modeling parameters and format types. Moreover, the 3D models should be transferred to distributed design scenes in time for an excellent user experience. Aiming at this purpose, we have investigated a platform of part library based on cloud. The architecture and workflow of the whole implementation in this paper is illustrated in Fig. 2.

In this scheme, a typical service process includes the following steps:

1. Receive the part type and corresponding parameter serial from the distributed clients all over the hosts or clients;
2. Create the 3D models according to the parameters in real-time by the ACIS modeler;
3. Transfer protosomatic .sat models into the requested formats for practical design applications through the InterOP API package suite;
4. Send the converted model files to clients via the network;

5. Import the models into the ongoing design environments.

Comparing with congener 3D part library, the framework proposed here contains two advantages. Firstly, the 3D models are created real-time on the server, which is much different from the method of preparing the 3D models beforehand. It is evident that a limited number of pre-created inventory parts cannot completely satisfy the diversiform needs from different users, as the combinations of parameter are nearly infinite. According to the part type and its parameters, the 3D modeling engine will create the 3D models, which can cover the combinations of parameters, avoid the massive modeling and save the storage space. Secondly, the clients in proposed framework can be directly embedded in the design environments. When 3D models are created and transferred onto local computers, which is transparent to users, they can be imported into the work sites and participated in the ongoing design. In addition, a pack of tools to aid the design processes, which is a beneficial supplement to the single connector. The connection or bridge between service provider on the cloud and the design envertiment on the user terminals is illustrated in Fig. 3, in which the kernel functional module is defined as PLS.

Figure 3 The Connection between Service Provider and Design Environment

Through the adapters to establish mutual connections between PLS provider and various kinds of host clients or systems, several modules such as 3D part modeling, transmission via network, preview in 3D browser and import in the real design environments can be achieved. The PLS provider is a comprehensive and detailed implementation, which is deployed on the cloud and takes nearly all the work that before the 3D models' transmission.

To realize the purpose, we have constructed the 3D modeling module without any APIs supported from general CAD software such as NX, Creo or CATIA to avoid the compulsive pre-requisite of their installations and all the functional modules are implemented in our owned way or from a lightweight API package.

The PLS provider can be divided into 3 subsystems: specification data including type/parameter selection, 3D model generation and PLS API package. The first subsystem is more well-known as part e-catalog, and the second one is for the uses of 3D part model generation in various formats according to the actual CAD design as it is implemented on a non-CAD environment. The exported functions of "service" is realized by PLS, a package of APIs in format of automation COM and Python interface, to be invoked by any other applications on certain criterions through the adapters, or connectors, which connect PLS provider with host systems, especially with the general CAD platforms such as NX, Creo, AutoCAD, CATIA, SolidWorks, etc. Moreover, the kernel module can also be wrapped and embedded in the website, which provides a feasible method for remote designers.

To our knowledge, the PLS method has currently not widely supported in the field of part library since most congeneric solutions are deployed in intrinsic environments, which limits the applications to a relatively narrow usage occasion.

4. Services In Part Library

Part library is an important auxiliary tool in mechanical or architectural design environments, and their detailed realization methods vary greatly, and there is no fixed pattern to follow and be obeyed. For a PaaS based large-scale part library for die and mold, it holds certain unique characteristic. In this section, a brief introduction to services provided in the part library is given.

4.1 Part Information Management

The primarily important issue of 3D part library is an effectual management of part information, in which concise classifications, indexes, searches and use guides are essential. In our implementation, the parts for die and mold are deeply investigated and more than 30000 parts are elaborately organized according to their shapes, usages, material and other attributes. The system provides various methods to locate the actual part, in which the most common way is by a three-level “category-type-parameter” navigator as shown in Fig. 4. The parts for die and mold are classified into dozens of categories, and when accessing one category, the exact needed part can be screened according to composable filters. Once the actual type name has been locked, their parameters can then be input or selected, along with which the corresponding 3D models can be created nearly at the same time.

Besides the three-level navigator, users can also access to the part through search by type, remark, keywords, etc. The library records the operating information including detailed parameters list, based on which some auxiliary modules such as part remark, design BOM, favorites folder or history record, etc, can be consequently established. These auxiliary functions give a means for the users to facilitate the practical design scenes.

Remark

a serial of tooltips on the information about the advantages/disadvantages, key points, user specifications, which are attached to an individual part. The remarks can append and edited by the remote users as a guide for further designs.

BOM: the part information about the products in a real design scene, the most important data in PDM/ERP information system. BOM also can be created by the design information in CAD systems through traversing the assembly trees. The information is stored in the database and can be exported into external file formats such as Excel .xls/.csv according to the pre-defined template, which is easily customized.

My favorites: the shortcuts to access the most commonly used parts with all parameters reserved in the personal account for each designer.

History records: the shortcuts to access the previously visited parts with all parameters reserved in the personal account for each designer.

With the above entrances, the users have the ability of agilely locating the needed parts from the enormous library. Meanwhile, these tools facilitate the design processes as they have been seamlessly embedded into the CAD software, and the users can therefore invoke the functions at any times.

3.2 3D Part Modeling and Format Conversion

Without the support from built-in CAD systems, real-time 3D creation and preview of the parts is a great challenge for the library. In our implementation, the 3D part models are created with ACIS, one of the most popular geometry engines. ACIS has a powerful ability of 3D modeling, supporting entity, wireframe, surface and other geometric elements. Through the flexible handle of these elements, the high-precision geometric model can be created. From the view point of efficiency, ACIS employs the object-oriented data structure and C++ is its development language, thereby, it holds an efficient operating and modeling performance. More important, it is an original and lightweight modeling with only dozens of dlls, so it is very convenient for deployment, especially on the web.

To ensure that the input parameters list will construct a valid geometric combination, a pre-check mechanism is embedded before 3D modeling. That is, although the geometric sizes of a body are in a complicated set of constraints and relations, the user will achieve a correct list however he inputs or selects parameters as the knowledge about dependency relations and rules have been inherent in the modeling. As a public part library, the system provides not only 3D models, but also 2D drawings. The parameters are input into the 3D modeler, in which .sat file are firstly created, and InterOP(for 3D) and PHLV5(for 2D) are then adopted to convert it into various formats including .stp, .iges, dxf, dwg, hsf, etc. The realtime preview for 3D models is realized based on Hoops with .hsf file. Figure 5 lists the formats that InterOP R24 supports.

The main challenge in our work is the huge modeling workload, as most 3D parts are created in native ACIS API for shape parameterization, though the same geometric features are assigned to share the common codes. Great flexibility in the deployment of modeling module and independence of CAD software are worthy of the cost.

3.3 Geometry and Rule Constraints

Because the geometric sizes in many parts are interrelated or even contain complex constraints, it is necessary to deeply investigate an effective way to represent and realize the relations. In this paper, we adopt various kinds of knowledge to handle the problems.

Firstly, the parameter linking tables are the uppermost method to store and handle the geometric relations. The parameters of a part are divided into active and passive ones, and if an active parameter has been changed, the ranges, values, or even precisions of its associated passive parameters will also be correspondingly changed. A string expression evaluator is developed and embedded to compute the current value, range and precision of each parameter. Figure 6 illustrates a parameter table for one part, in which all parameters are associated.

Since the ranges of the parameters are stored in the form of strings, the expression evaluator is to handle them in an analytic method as in code compilation. All the information about the parameters is being updated during their inputting process. When the users input a value, the system will also check its compliances either in range or in precision. In this way, the part library will realize the ability of pre-check of parameters instead of post-check, that is, users will not wait until the 3D modeler returns the failure information to find out the combination of parameters is incorrect. The post-check from 3D modeling will face with the judgment of geometric singularities, and therefore is incompletely reliable together with time-consuming. The pre-check in our system help avoid the repeated this kind of trials and errors. For particularly complex 3D model, the relationships among parameter chain are nesting or even circular referenced, and on these occasions a partial order of parameters for minimum influences should be built by altering certain prior restraints to hind validations.

Second, the rules of parameter and alternation selection (as shown in Fig. 7) are also stored in the knowledge database with an editing tool(KDT, Knowledge Database Tool) for users with the format of "IF-THEN" expressions. The base parameter setting can also be referenced in Fig. 4. KDT supports an active monitoring for the change of parameters or processing alternation, with which if their inputs or selections trigger certain rules, the later will automatically come into force and meanwhile KDT will take corresponding per-defined measures to insure the data correction.

The KDT is divided into knowledge database module and knowledge reasoning module based on Protégé[31], an open source ontology-based knowledge acquisition, management and deployment tool developed by Stanford University. The knowledge database loads ontologies of geometric constraints and reasoning rules constructed from Protégé, and stores them in OWL structures and documents with RDF/XML format. The reasoning module parses the querying statements, checks the satisfaction and inclusion of instance, rule and their knowledge object set, and then constructs a searching tree through reverse chain reasoning algorithm, to finally complete the retrieval, matching and reasoning of the geometric and semantic information.

Third, feedback edit and associated assembly are two attractive and peculiar operational modes in our implementation, which provide a mechanism of sending the current design information back to the PLS provider, and assist it to automatically screen and restore suitable parameter lists for 3D modeler. On these occasions, PLS is more like a knowledge receiver and convertor from the real designs. For example, a part has been firstly assembled into the practical design. If the designer intends to change its specifications, with the help of PLS provider the only necessary action is to select the 3D model and return back to the interface of parameter list by a pop-up menu, the original assembly information (position, constraint, keep-space and cut) will continue to be applied to the alternate parts. Figure 8 is an example that the old part "LRBS60-10" is replaced by the new one "LRBS100-15" without repeating the assembling process, which saves much time.

The data relevance and knowledge in PLS is not only for the parameters of one part, but for the "successive assembly". It is well known that some parts are linked with one or more another, the later are called as “associated parts”, so if it is imported into the design, it is very likely that its “associated parts” will also be added to the assembly. For example, once the type and specifications of the mold base are decided, the assembling position of the hexagon bolt with other standard parts are therefore fixed. For such kinds of standard parts, it is not necessary to manually operate one by one. Successive assembly can help the design automation aiming at this occasion. When a mold base is initialized, the required parts and their corresponding assembling position are queried from the associated tables, and then homogeneous matrices are computed, by which the successive parts will be also automatically imported and assembled onto the mold base. The knowledge in successive assembly is solidified in an associated table, which help reduce the repeated operations and manual errors, therefore will greatly enhance the effectiveness of large-scale assembly design.

5. Service Delivery Based On Cloud

The service delivery based on the cloud includes two aspects of data processing as mentioned in Sect. 2.1 as namely: ‘Integration of Distributed Resources(IoDR)” and “Distribution of Integrated Resources(DoIR)”.

IoDR provides a mechanism that uniformly manages the basic infrastructure(including software, hardware and cloud) and data(including parameter setting, 3D models and other related resources of the part library distributed in different sites even all over the servers). IoDR wraps and integrates them into an integral PLS provider within a PaaS scheme, which is thereupon transparent to their users such as PLS adapters, PLS hosts. The later thus need no longer cares about any information on the category, location and ownership of specific resource. Therefore, the distributed resources on the cloud perform in the form of centralization.

DoIR provides another mechanism for the distributed clients or PLS hosts to effectually retrieve the services from the identical PLS provider. The PaaS concept embedded in PLS adapters connects PLS provider with external applications in an interoperable bi-directional channel. The main work in this issue is to import PLS API package into the local platform. Therefore, the PLS adapters bring the two sides together and bridge the gaps between them by exchanging data and services in a common module. For instance, CAA(Component Application Architecture) can be employed as the connector for PLS provider and CATIA, the same as objectArx for AutoCAD, etc.

The implementation of DoIR is diverse according to different client applications while relatively simple compared with IoDR, therefore in this scheme the most important issue is on the later.

5.1 Basic Concept of IoDR

IoDR is designed to be composed of four layers, namely, resource layer, integration layer, service package layer and application interface layer. The resource layer encompasses the part library data and models required in PLS provider, which are deposited in distributed sites for different parts suppliers. These resources include structured data such as part specifications(size/constraints, material, accuracy, roughness, surface treatment, etc) and unstructured data such as part photos/images, 2D drawings, 3D models, attached documents, etc. The former is stored in the database while the later are saved as files in the cloud. The resource layer provides the basic management and accessing methods of the original part information. The integration layer is to wrap the distributed and heterogeneous resources into a unified part pool, which is transparency to their clients. With the unified part pool, the services can be packed on the cloud to provide for clients or users all over the Internet in the scheme of PaaS. Through the PLS adapter, the hosts can access all the services by importing the APIs into their own frameworks. The architecture of the IoDR is illustrated in Fig. 9.

5.2 Resource Pool Virtualization and Integration

Resources in the broad sense includes data about part library and computing ability, and since the latter has been deeply investigated in general fields, we will emphasize on the integration methods of part resources, which include structured and unstructured data as mentioned above. Moreover, they are often scattered stored as the related parts may be produced and manufactured by different suppliers. To keep respective data independence and confidentiality, the resources are arranged to be maintained by their owners. Therefore, a background program is developed for resource providers, supporting the full edit of their own resources in the part library.

Data virtualization improves the dynamic adaptability, centralizes the stored resources in a “pool” with large capacity, which does not emphasize on specific storage devices and locations any longer for their respective owners and users. The integrated management of resource pool realizes the adjustment of storage system and data movement without interrupting the current applications, which facilitates the deployment of part library on the cloud.

With the unified API package, part service can be achieved without knowing the specific details of the remote resource pool, which provides a transparent accessing method to all clients. Meanwhile, the clients will also receive indifferent services from PLS provider. By means of pool virtualization and integration, the distributed part resources are packed and exhibited as a whole, and the accessing control is achieved by differentiating registered user-types.

5.3 Deployment on the Cloud

After resource integration and function encapsulation, PLS provider will be deployed on the cloud for the access all over the web. In our implementation, a series of classes and functions are exposed to the external and PaaS scheme is responsible for their distribution. For example, Fig. 10 is a package of APIs in C/C++, with the help of automation technology, PLS adapters can import and invoke them in almost all programming languages.

The resource deployed in the cloud includes data, programs and virtual computer devices, which constructs a complete run-time environment for part library. The classes and APIs are hosted in an API gateway, which aims at providing a unique entrance for remote part library and is adopted for routing, composition and protocol transformation to handle the requests from clients. API gateway also help to protect the architecture from external attacks and also coordinate the balance of workloads. Through this way, the PLS provider can be safely imported to different hosts.

We have constructed a PaaS framework for PLS provider with ACS(Azure Container Service), a container hosting environment, which is be used for container deployment and management in cloud scheme. For an effectual cloud service, communication optimization, security configuration and load balance are also taken into depth consideration. After stress testing, 3000 requests can be concurrent without significant user latency as the preliminary construction, which will be further extended for the usage in more large-scale applications.

6. Services In Cad Platform

With the kernel functions of PLS provider and various hosts/clients, we have implemented a PaaS deploying framework on cloud. The PLS adapters are developed according to the hosts' development toolkit as they serve as plug-in programs in the hosts for the connection between themselves and PLS provider. Through different kinds of PLS adapters, the hosts have the ability of invoking the functions exported from the PLS provider, and CAD systems are the most common hosts. As PLS adapters are hosted in CAD environments, it is reasonable for them to be developed with API package of the exact host, importing the remote PLS provider meanwhile. Figure 11 is three typical hosts as AutoCAD, Catia and NX. The PLS adapter for AutoCAD is developed with ObjectArx, that for Catia is with CAA (Component Application Architecture) and that for NX is with NXOpen.

The customized menus in the CAD systems provide an access to visit the PLS provider on cloud, to obtain the part model and information and to insert the 3D model into ongoing design. Through this method, the remote resources are flexibly linked with local environments.

7. Conclusion

In this paper, a framework of part library on cloud is investigated and developed, in which the part library is encapsulated as the form of "Service". The service is divided into three section, namely: PLS provider, PLS adapter and PLS host. All the part data services are available in PLS provider through a package of API, the PLS hosts receive the part library service from PLS provider with the PLS adapter as the bridges. In this scheme, remote users can achieve all part information without installing a huge library in his own computer, instead, a lightweight PLS adapter is adequate.

The part library on cloud proposed in this paper has been successfully deployed and used in many manufacturing enterprises and brings great benefits to them. Compared with the traditional standard part library system, this system holds the following characteristics:

1) Cross-platform: The framework doesn't depend on a specific CAD platform. The part library is separated from CAD operating modules, and an independent 3D modeling kernel(ACIS) is embedded in

the system. Afterwards, a rapid expansion and application in different CAD platforms are realized to avoid developing several specific systems for different CAD platforms.

2) Cloud sharing: A unified cloud-based part library service is developed to facilitate an integrated management and update of enterprise data and knowledge, which is conducive to teamwork.

3) Convenient operations: Providing the uniform interface, the PLS can be easily and seamlessly integrated in various hosts including most CAD systems as part of the design processes without any differences. The library is designed for die and mold design process according to the user's operating habits, which greatly enhances the user's design efficiency. Moreover, the editing engagement in CAD systems can be fed back to PLS.

The framework holds some shortcomings that needed further improved. Firstly, the 3D models are created with ACIS and converted into .stp or .iges, which are not the native format for a CAD platform. Therefore, the imported parts only contain information with geometry and attributes, while its features and structures are missing, which brings the inconvenience for edit. Direct driving of native model in CAD systems is an interest in our later research. Another, the types in part library cover only die and mold currently, other kind of parts should also be taken into consideration. As for the further work, we plan to extend the part library from die and mold to more kinds of parts such as factory automation in manufacturing industry.

Declarations

Ethical Approval: There are no ethical problems as this research belongs to the field of engineering technology, and does not involve any human body related data.

Consent to Participate: All authors consent to participate.

Consent to Publish: All authors consent to publish.

Authors' contributions: All authors cooperatively developed the software in past several years.

Funding: Not applicable

Conflicts of interest/Competing interests: The authors declared that they have no conflicts of interest or competing interests to this work.

Code availability (software application or custom code): The software application has been used in many die and mold enterprises.

Availability of data and material (data transparency): The data and material has been used in many die and mold enterprises.

References

1. Qingming F, Liu G, Wang W (2011) Development of Die Sets Standard Parts Library Based on Pro/E. Procedia Eng 15:3802–3807
2. Jong W, Wu C-H, Li M (2011) Web-Based Navigating System for Conceptual Mould Design with Knowledge Management. Int J Prod Res 49(2):553–567
3. Lee J-H, Kim S-H, Lee K (2012) Integration of Evolutional BOMs for Design of Ship Outfitting Equipment. Comput Aided Des 44(3):253–273
4. Ma Y, Tor S-B, Graeme ABritton (2003) The Development of a Standard Component Library for Plastic Injection Mould Design Using an Object-Oriented Approach. The International Journal of Advanced Manufacturing Technology 22(9–10):611–618
5. Mok H-S, Kim C-H, Kim C-B (2011) Automation of Mold Designs with The Reuse of Standard Parts. Expert Syst Appl 38(10):12537–12547
6. Huang B, Li C, Yin C, etc. Cloud manufacturing service platform for small- and medium-sized enterprises. The International Journal of Advanced Manufacturing Technology, 2013, 65: 1261–1272
7. Xun Xu. From Cloud Computing to Cloud Manufacturing. Robotics and Computer-Integrated Manufacturing. 2012, (28): 75–86
8. Gao R, Wang L, R.Teti, etc. Cloud-Enabled Prognosis for Manufacturing. CIRP Annals-Manufacturing Technology, 2015, 64, 749–772
9. Brecher C, Lohse W (2009) M.Vitr. Module-based Platform for Seamless Interoperable CAD-CAM-CNC Planning. Advanced Design and Manufacturing Based on STEP. Springer, London
10. C.Brecher,MVitr, JWolf (2006) Closed-Loop CAPP/CAM/CNC Process Chain Based on STEP and STEP-NC Inspection Tasks. Int J Comput Integr Manuf 19(6):570–580
11. V.Velde PJMC. Runtime Configurable Systems for Computational Fluid Dynamics Simulations. PhD Thesis. Auckland: Department of Mechanical Engineering, University of Auckland, 2009
12. Nassehi A, Newman ST (2008) XW.Xu, etc. Toward Interoperable CNC Manufacturing. Int J Comput Integr Manuf 21(2):222–230
13. Mokhtar A (2010) M.Houshmand. Introducing a Road Map to Implement the Universal Manufacturing Platform Using Axiomatic Design Theory. Int J Manuf Res 5(2):252–269
14. SH.Suh SJ.Shin JS (2008) .Yoon, etc. A New Paradigm for Product Design and Manufacturing Via Ubiquitous Computing Technology. Int J Comput Integr Manuf 21(5):540–549
15. BE.Lee (2009) SH.Suh. An Architecture for Ubiquitous Product Life Cycle Support System and Its Extension to Machine Tools with Product Data Model. Int J Adv Manuf Technol 42(5):606–620
16. Wang X (2012) X.Xu. An Interoperable Solution for Software Integration and Product Data Exchange. Enterprise Information Systems 6(3):291–314
17. Feng Han T, Su (2005) Zong-ying Ou, etc. Development of Standard Part Library for Three-Dimensional Stamping Die by Using Solidworks. Journal of Dalian Institute of Light Industry

24(3):211–214

18. Zhang Z, Zhu S (2003) Qingchun Zheng, etc. The Secondary Development of SolidWorks by VC++. Journal of Tianjin Institute of Technology 19(4):69–71
19. Jiande Wang. Research on Feature Based Information Model in Part Library and Its Implementation. Dissertation, Harbin Engineering University, 2006
20. Lu Y, Liao WH (2004) X.Huang, etc. Research on Data Description Norm and Building Techniques of Web-Based Parts Library. China Mechanical Engineering 15(11):987–990
21. XR.Xu. Research on Development and Application of Distributed Library System Based on Web Service. Dissertation, Zhejiang University, 2006
22. JH.Zhou. Research and Implementation of Part Data Model Library Management System. Dissertation, Zhejiang University, 2005
23. Joonmyun Cho S, Han H, Kim (2006) Meta-Ontology for Automated Information Integration of Parts Libraries. Comput Aided Des 38(7):713–725
24. Cho J, Kim H, Han S (2006) Content-Oriented Knowledge Modeling for Automated Parts Library Ontology Merging. Lect Notes Comput Sci 3865:324–333
25. Jin B, Teng HF (2008) TS.Wang, etc. Product Design Reuse with Parts Libraries and An Engineering Semantic Web for Small- And Medium-Sized Manufacturing Enterprises. Int J Adv Manuf Technol 38(11):1075–1084
26. http://www.3dcontentcentral.com/
27. http://www.cadclick.de
28. http://www.clearpathmedia.com/InPart_Web_Mirror
29. http://www.traceparts.com/
30. http://www.web2cad.co.jp/
31. https://protege.stanford.edu/

Figures

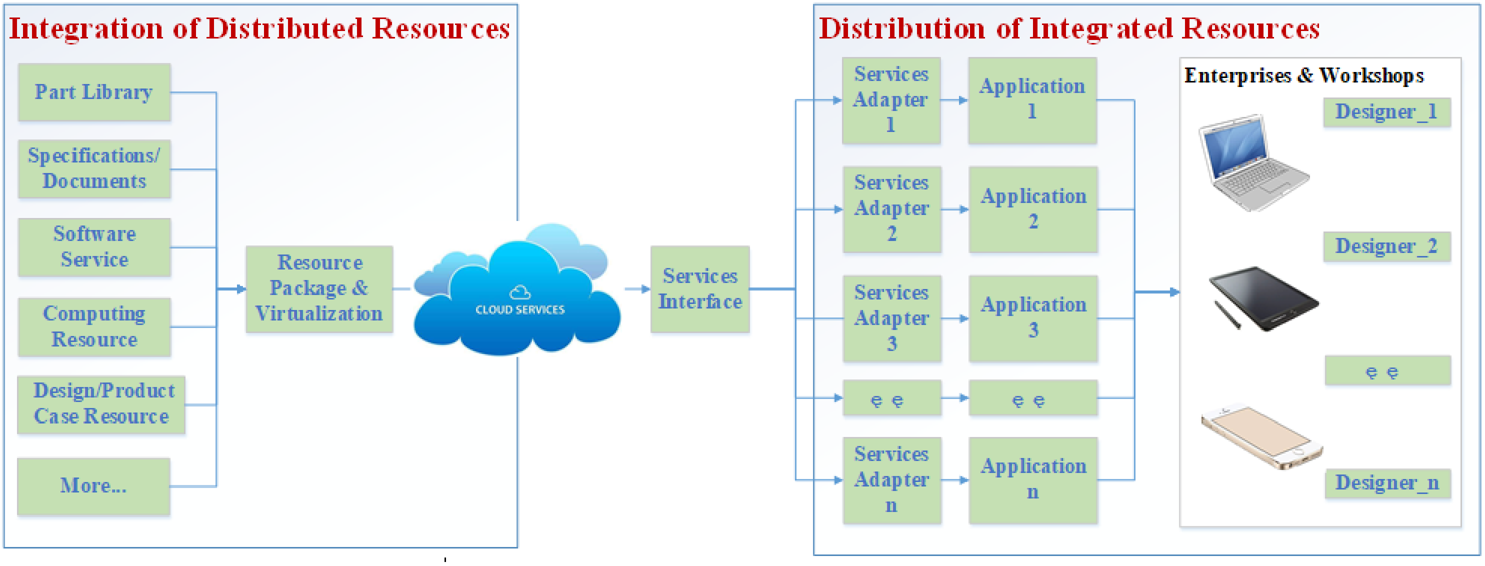


Figure 1

The Structure of the Two Concepts

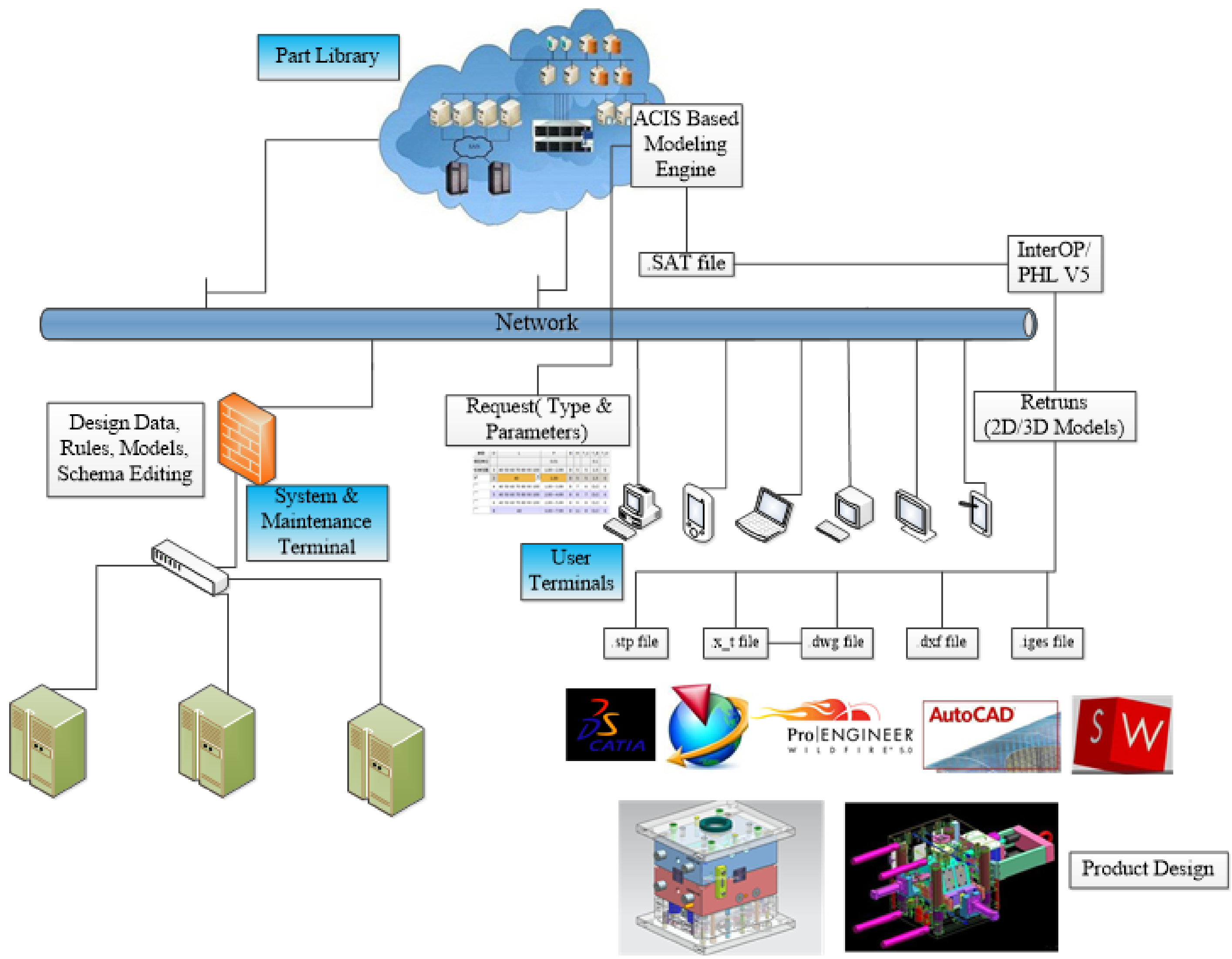


Figure 2

Architecture & Workflow of Part Library System on Cloud

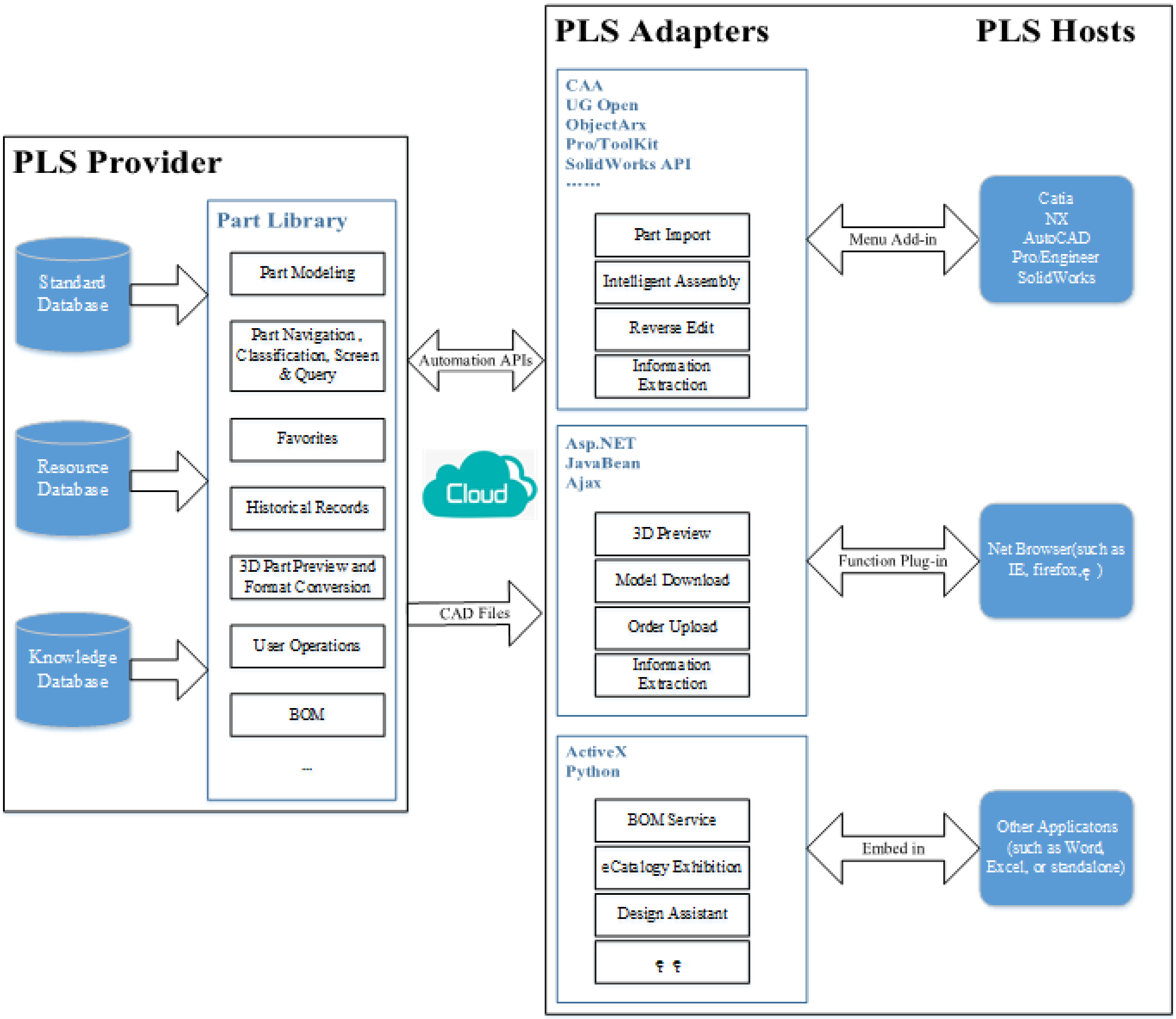


Figure 3

The Connection between Service Provider and Design Environment

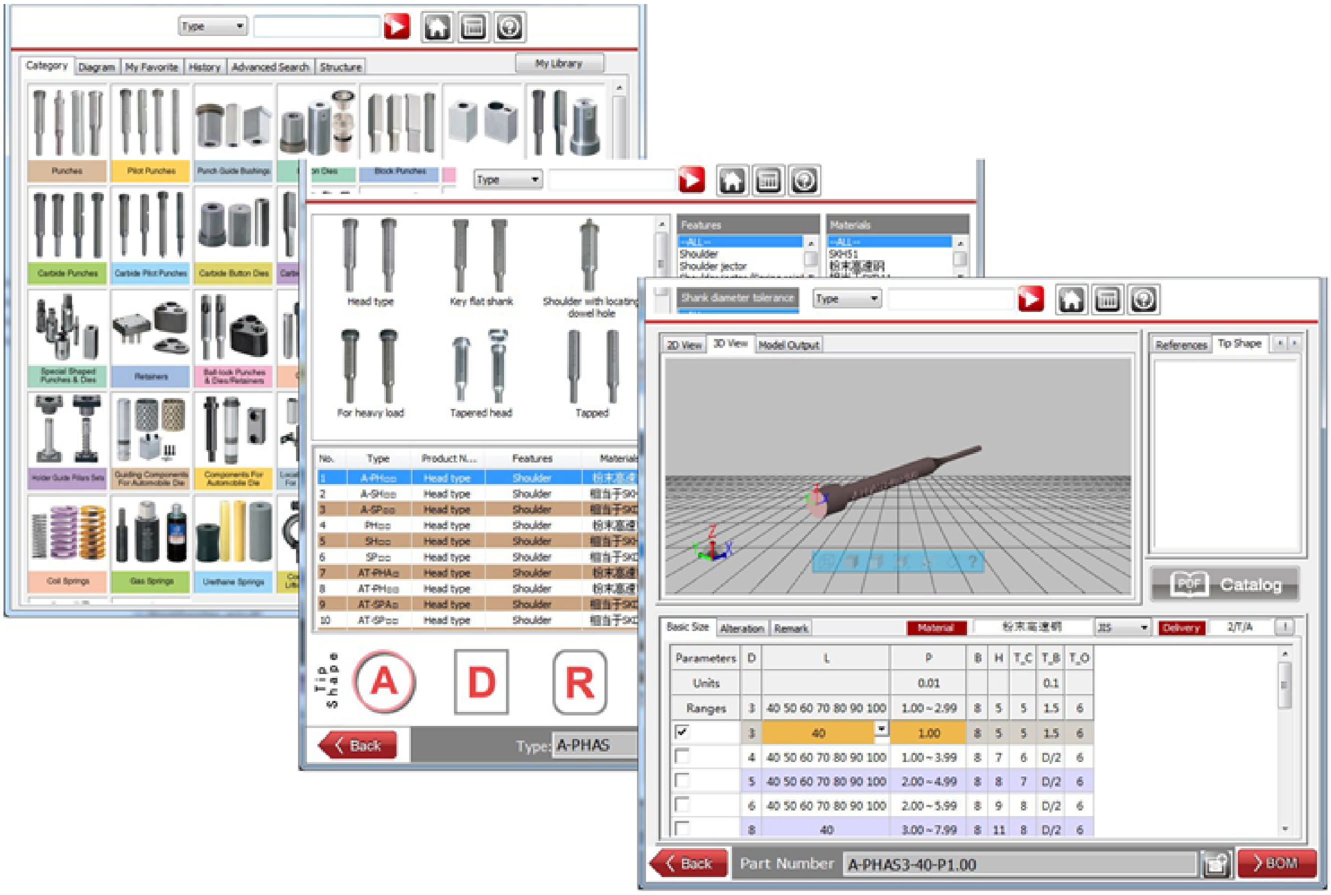


Figure 4

The Three-level Navigator

Software		Version	Graphics Data	Precise Geometry	Semantics PMI
V4		4.1.9 - 4.2.4	4.1.9-4.2.4(Generated from Geometry)	4.1.9 - 4.2.4	N/A
V5		R6 - R23 (V6-5 R2013)	R8 - R23 (V6-5 R2013)	R6 - R23 (V6-5 R2013)	R6 - R23 (V6-5 R2013)
3DXML		v4.3	v4.3		
Creo		16-Creo 2.0	WF3 - Creo 2.0 16 - WF2(Generated from Geometry)	16 - Creo 2.0	16 - Creo 2.0
NX	NX - PS	11 - NX8.5	NX6 - NX8.5 11 – NX6 (Generated from Geometry)	11 - NX8.5	11 - NX8.5
	NX Direct	NX1 – NX8.5	NX6 - NX8.5 NX1 – NX6 (Generated from Geometry)	NX1 – NX8.5	NX1 – NX8.5
SolidEdge	SE - PS	V18 – ST5	ST– ST5 V18 – V20 (Generated from Geometry)	V18 – ST5	N/A
	SE Direct	V18 – ST5	ST– ST5 V18 – V20 (Generated from Geometry)	V18 – ST5	
SW	SW - PS	98 - 2013	98 - 2013	98 – 2013	N/A
	SW Direct	2003 – 2013	2003 - 2013	2003 – 2013	
Inventor		V11 – 2014	v7-2014 v6(Generated from Geometry)	V6 - v2014	N/A
Parasolid	PS - PS	10.0 - 25.0	10.0 - 26.0 (Generated from Geometry)	10.0 - 26.0	N/A
	PS Direct	14 – 26	14 – 26 (Generated from Geometry)	14 – 26	
STEP		203, 214	203, 214 (Generated from Geometry)	203, 214	N/A
IGES		Upto 5.3	Upto 5.3 (Generated from Geometry)	Up to 5.3	N/A
VDA-FS		1.0 - 2.0	1.0 - 2.0 (Generated from Geometry)	1.0 - 2.0	N/A
ACIS		R1 - R24	R1- R23 (Generated from Geometry)	R1 - R24	N/A

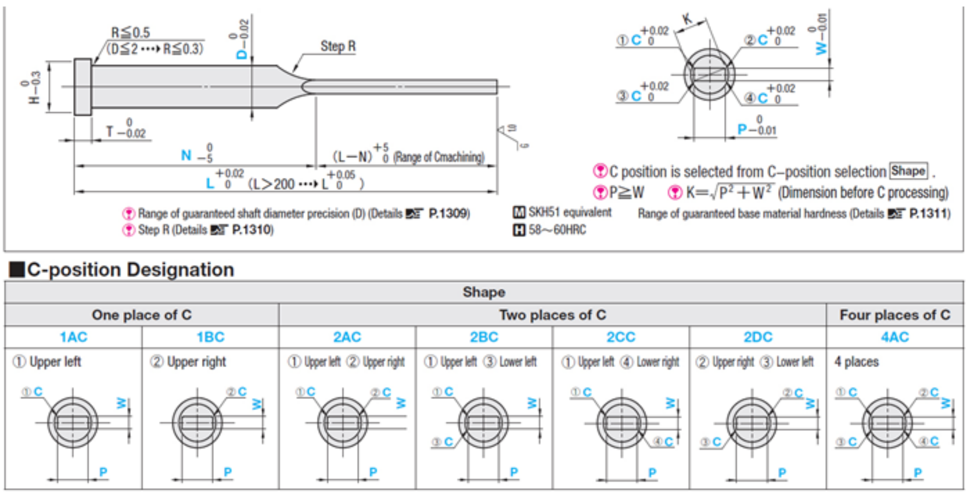
Figure 5

The formats InterOp R24 supports

D	L	P-0.01	W-0.01	A-0.01	B-0.01	R-0.01	S-0.01	Kmax	Wmin	bb	B1	dd	F	K_check-0.01	T_O	T_C	T_B-1
10	16 20 22 25 30 35 40	2.82 ~ 6.00	2.01 ~ !P	1.01 ~ W-1	1 ~ !A	0.2 ~ 40	0.2 ~ !(P-W)/2-R,40	6.00	2.00	10	6	6.4	6	sqrt((P-2*R)*(P-2*R)+(A-2*R)*(A-2*R))+2*R ~ Kmax	1	1	D/2
13	16 20 22 25 30 35 40	2.82 ~ 8.00	2.01 ~ !P	1.01 ~ W-1	1 ~ !A	0.2 ~ 40	0.2 ~ !(P-W)/2-R,40	8.00	2.00	10	6	8.4	7.5	sqrt((P-2*R)*(P-2*R)+(A-2*R)*(A-2*R))+2*R ~ Kmax	1	1	D/2
16	16 20 22 25 30 35 40	3.31 ~ 10.00	Wmin ~ !P	1.01 ~ W-1	1 ~ !A	0.2 ~ 40	0.2 ~ !(P-W)/2-R,40	10.00	2.50	10	6	10.6	8	sqrt((P-2*R)*(P-2*R)+(A-2*R)*(A-2*R))+2*R ~ Kmax	1	1	D/2
20	16 20 22 25 30 35 40	3.82 ~ 12.00	Wmin ~ !P	1.01 ~ W-1	1 ~ !A	0.2 ~ 40	0.2 ~ !(P-W)/2-R,40	12.00	3.00	12	8	12.6	10	sqrt((P-2*R)*(P-2*R)+(A-2*R)*(A-2*R))+2*R ~ Kmax	1	1	D/2
25	16 20 22 25 30 35 40	4.82 ~ 16.00	Wmin ~ !P	1.01 ~ W-1	1 ~ !A	0.2 ~ 40	0.2 ~ !(P-W)/2-R,40	16.00	4.00	12	8	16.6	12.5	sqrt((P-2*R)*(P-2*R)+(A-2*R)*(A-2*R))+2*R ~ Kmax	1	1	D/2
32	16 20 22 25 30 35 40	5.82 ~ 20.00	Wmin ~ !P	1.01 ~ W-1	1 ~ !A	0.2 ~ 40	0.2 ~ !(P-W)/2-R,40	20.00	5.00	15	10	20.6	16	sqrt((P-2*R)*(P-2*R)+(A-2*R)*(A-2*R))+2*R ~ Kmax	1	1	D/2
38	16 20 22 25 30 35 40	6.82 ~ 26.00	Wmin ~ !P	1.01 ~ W-1	1 ~ !A	0.2 ~ 40	0.2 ~ !(P-W)/2-R,40	26.00	6.00	15	10	26.6	19	sqrt((P-2*R)*(P-2*R)+(A-2*R)*(A-2*R))+2*R ~ Kmax	1	1	D/2
45	20 22 25 30 35 40	6.82 ~ 35.00	Wmin ~ !P	1.01 ~ W-1	1 ~ !A	0.2 ~ 40	0.2 ~ !(P-W)/2-R,40	35.00	6.00	20	14	36.0	22.5	sqrt((P-2*R)*(P-2*R)+(A-2*R)*(A-2*R))+2*R ~ Kmax	1	1	D/2
50	20 22 25 30 35 40	7.82 ~ 40.00	Wmin ~ !P	1.01 ~ W-1	1 ~ !A	0.2 ~ 40	0.2 ~ !(P-W)/2-R,40	40.00	7.00	20	14	41.0	25	sqrt((P-2*R)*(P-2*R)+(A-2*R)*(A-2*R))+2*R ~ Kmax	1	1	D/2
56	20 22 25 30 35 40	8.82 ~ 45.00	Wmin ~ !P	1.01 ~ W-1	1 ~ !A	0.2 ~ 40	0.2 ~ !(P-W)/2-R,40	45.00	8.00	20	14	46.0	28	sqrt((P-2*R)*(P-2*R)+(A-2*R)*(A-2*R))+2*R ~ Kmax	1	1	D/2

Figure 6

Parameter Table for a Part



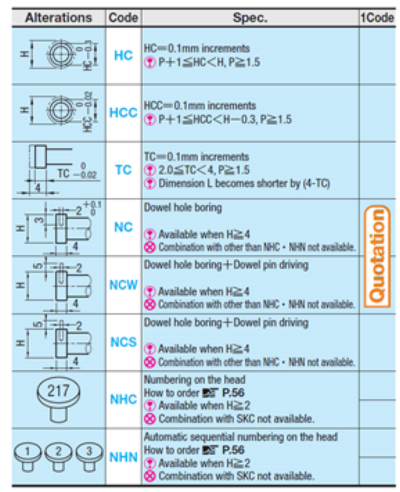


Figure 7

Parameter and alternation rules

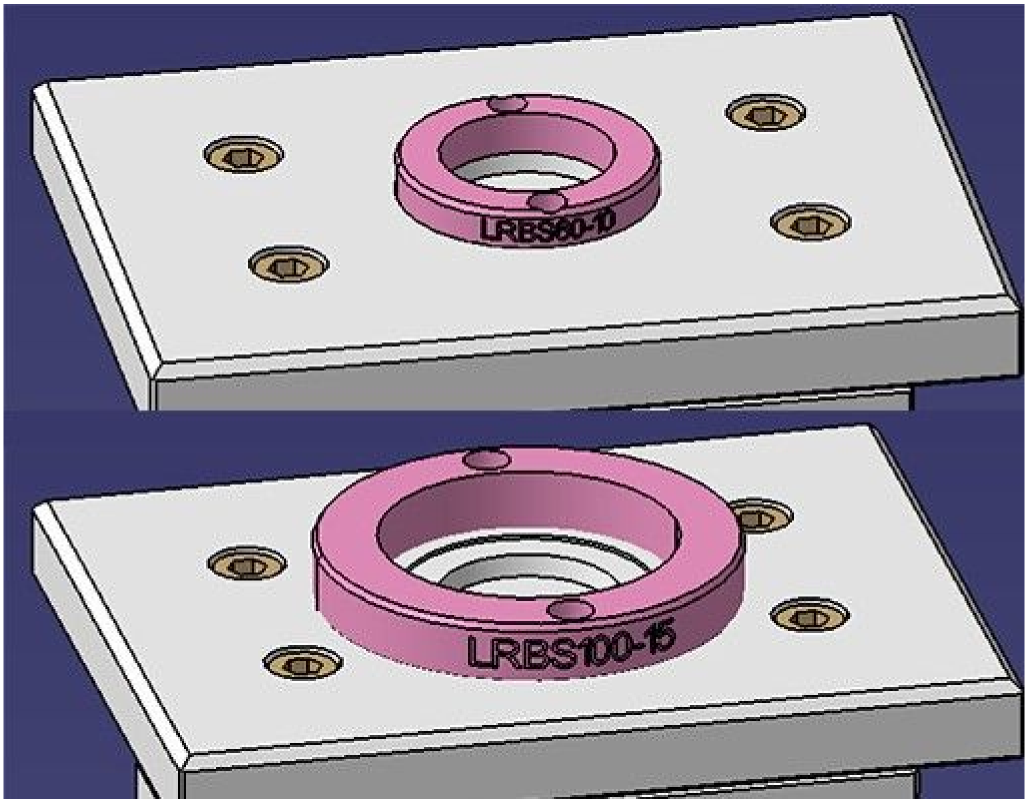


Figure 8

Information Reservation in Backward Edit

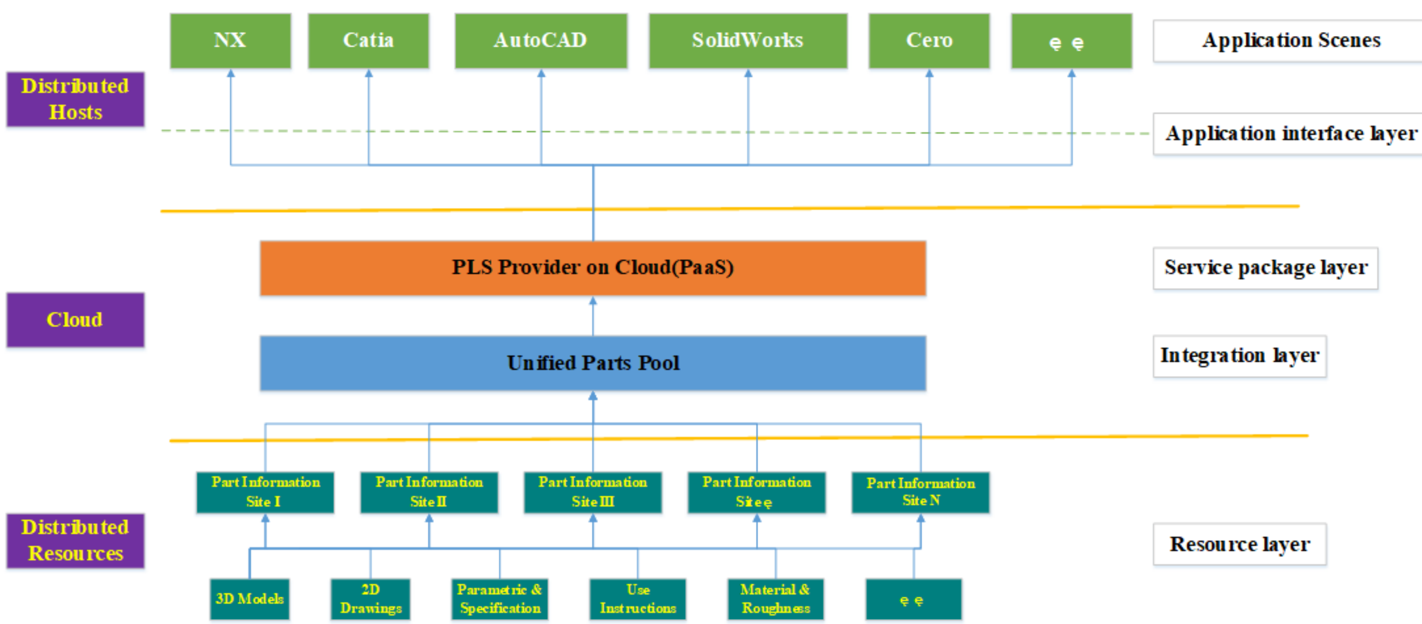


Figure 9

The Architecture of the IoDR

```
extern _declspec(dllexport) int ShowInterfaceDlg1(int i);
extern _declspec(dllexport) int ShowInterfaceDlg2(int index,BOOL C_Value,int Setting);
extern _declspec(dllexport) int ShowInterfaceDlg3(char* strPara);
extern _declspec(dllexport) int GetParaCount();
extern _declspec(dllexport) char* GetParaName(int index);
extern _declspec(dllexport) char* GetParaVlaue(int index);
extern _declspec(dllexport) char* GetTypeName();
extern _declspec(dllexport) char* GetOrder();
extern _declspec(dllexport) void ClearBom();
extern _declspec(dllexport) void ClearAlterBom();
extern _declspec(dllexport) void AddToBom(char* cPara);
extern _declspec(dllexport) void AddToAlterBom(char* cPara);
extern _declspec(dllexport) void ShowBom();
extern _declspec(dllexport) void ShowAlterBom();
extern _declspec(dllexport) void Init(int i);
extern _declspec(dllexport) void Uninit();
extern _declspec(dllexport) int GetExeState();
extern _declspec(dllexport) void ClearData();
extern _declspec(dllexport) void Counters(int index);
extern _declspec(dllexport) void SetRestart();
```

Figure 10

A Package of Functions from PLS Provider

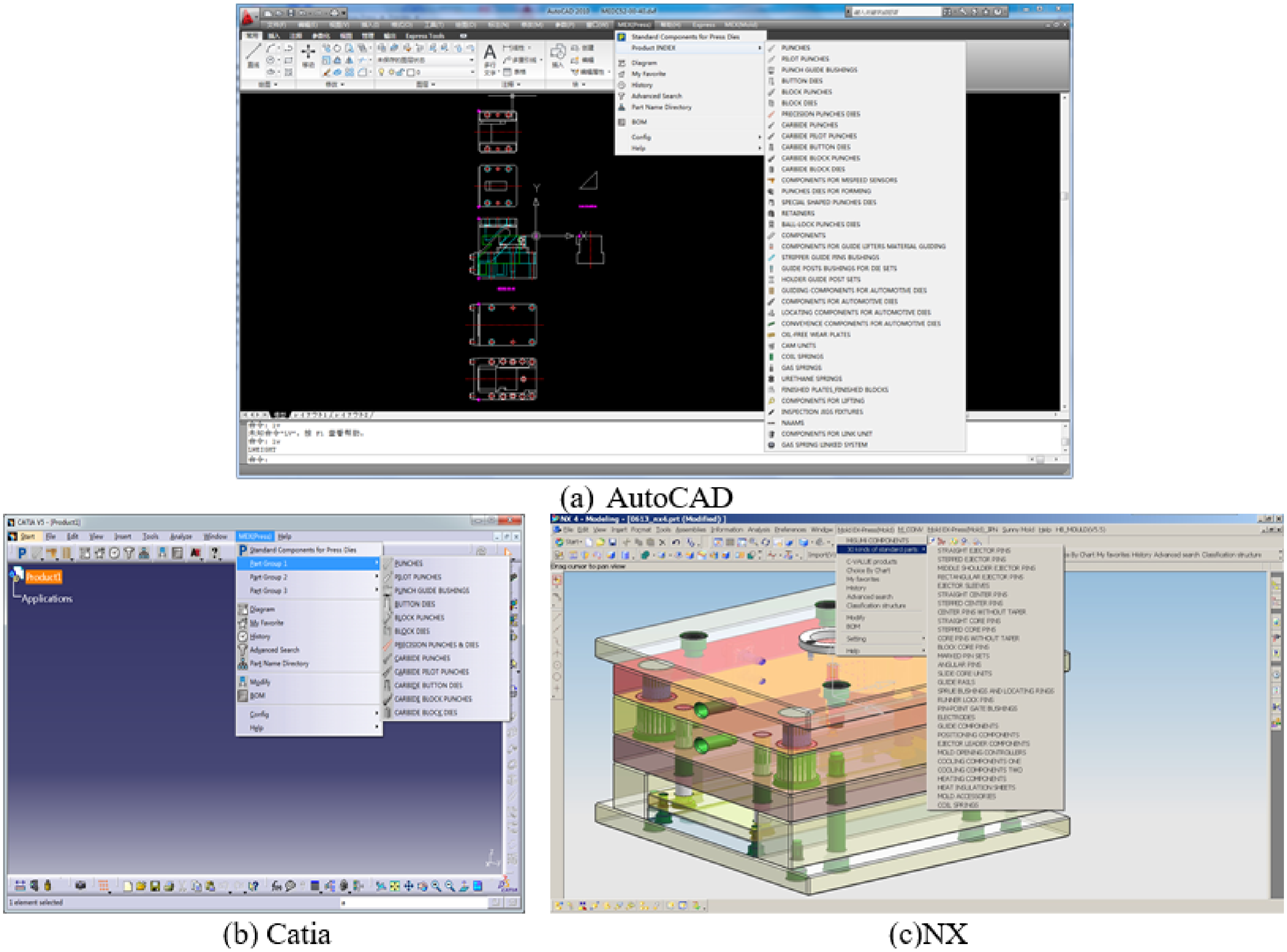


Figure 11

PLS Adapter Hosted in CAD System

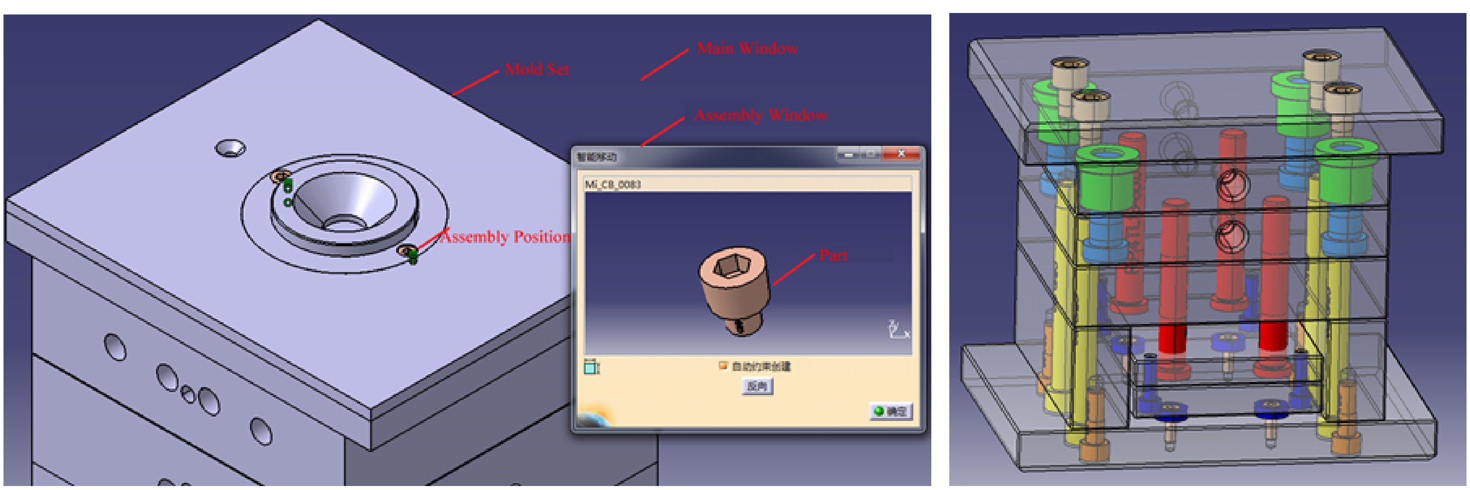


Figure 12

Mold Set Design with PLS System in Catia