Towards ubiquitous data access service of 3D part library in cloud scheme with host independence

Wei Liu · Chuipin Kong · Qiang Niu · Xionghui Zhou

Received: 2 February 2021 / Accepted: 28 March 2022 / Published online: 2 April 2022
© The Author(s), under exclusive licence to Springer-Verlag London Ltd., part of Springer Nature 2022

Abstract

Most current part libraries are developed and deployed in specific usage scenario or CAD systems, which inevitably brings obstacles in information sharing and exchanging of parts. To reduce repeated development and provide a uniform platform 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 modeler with native ACIS, and then, they are converted into 3D files in different formats for practical needs; finally, these models are imported into CAD systems or other platforms in real designs. The framework is composed of three components, namely, as PLS (part library service) provider, PLS adapters, and PLS 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 an embedded module or run on the websites or even mobile terminals. The PLS provider is deployed and maintained on cloud, and designers can acquire remote part information within a local ongoing project. In the detailed construction of this part library, diversiform knowledge in part parameters and structures is implanted to define the geometry and rule constraints for 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. This concept has been implemented within a PaaS framework to provide the ubiquitous 3D part data accessing route, which has been successfully applied in a large number of enterprises, and accumulates considerable practical cases.

Keywords PLS · Cloud design · Cross-platform · 3D modeling · Host independence

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 mechanical and structural design by providing necessary predefined parts, which helps to improve their efficiency. It is therefore a good concept 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 quantity and series can hardly completely satisfy various designers’ increasing requirements.

To expediently obtain 3D models for agile design, certain e-catalogues or 3D part libraries have appeared to access the 3D part information in recent years, and a considerable proportion of them were constructed by part suppliers to promote their commodities. In addition, several enterprises also developed individual modules or subsystems in their existing PDM/ERP platforms to realize effective management of the parts information from different suppliers [1–5].

From the perspective of system deployment, part libraries are generally constructed on a single CAD system to completely exploit the performance of built-in APIs or intrinsic modules. Thus, a lot of extra work can be economized during the developing periods. However, the libraries will inevitably rely heavily on the exact host CAD platform and
cannot be conveniently transplanted to another one. Therefore, these kinds of part libraries are hard to be extended, transplanted, exchanged, or shared by other related applications in the phase of upstream or downstream partners. As a result, the individual part library is enclosed to form an isolated island and does not contain the ability of communicating information or exchanging data with each other. Thus, information redundancy, mutual contradiction, or repeated developments are the frequent troubles both for designers or for enterprises. Moreover, they rarely support the concept and mechanism of providing their part service alone for designers in remote workplace, that is, to exclude the predetermined supports from their host CAD platform and provide “service” for more users. 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 will be deployed on the remote servers in a one-off pattern and provide ubiquitous data access services for the dispersed designers synchronously.

With the scheme of cloud, the basic functions of part library are configured in the server, and on each client, an adapter also needs to 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 are 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 predefined 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 CAD 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, the InterOP APIs are adopted to transfer them into various formats, such as stp, igs, x_t, and hsf, for 3D designs on different CAD platforms, realizing the concept of “Created once, used almost everywhere,” which accords with the original intention of cloud, especially of PaaS.

In addition, to effectively manage the parameters in an exact definition of 3D models, the methods of how to seamlessly and efficiently embed design knowledge into our part library is also investigated, with which the data in PLS provider can be easily edited or customized in line with practical needs. Flexible reconfigurable and effective representation in the embedding of diversiform knowledge is an important advantage in the reuse of existing geometrical, structural, or logical information in the stage of part information edit, which is an emphasis in our blueprint.

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 participant in the real design from remote server instead of only providing model download with real-time 3D modeling generation. The rest of this paper is organized as follows. In next section, we will review the latest advancements in related fields. The architecture and method will be presented in Sect. 3, and detailed technologies in our 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 well adapt the new situation of distributed and rapid design patterns, and the demand-sides will seek and obtain their needs from all over the world. Cloud scheme supports the feasibility and implements the basic infrastructures for data and service sharing from 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 for 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 described 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. The deployment and physical distributions are transparent and exhibit as a whole as for their 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.

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 include physical categories and functional categories (including software). And after IoDR and package, all available backstage resources are transferred into service 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 et al. [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. Nassehi et al. [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 strategy, different components of CAD/CAM/CNC chain can exchange information with another regardless of the native standards. Mokhtar and Houshmand [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

---

**Fig. 1** The structure of the two concepts
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 included 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 and Xu [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. The establishment of part library is to realize the management and sharing of part information, requiring its universality and openness. The German Institute for Standardization (DIN) had formulated a series of standards such as DIN4000-XX, DIN4001, and DIN66304. Afterwards, the European Organization for standardization (CEN/CENELEC) and the International Organization for Standardization (ISO) jointly formulated the international standard ISO13584: Part-LIB on the basis of German standards to meet the requirement of manufacturing globalization. ISO13584 absorbs the basic idea of DIN and is also influenced by STEP standard, introducing the formal description method of EXPRESS language.

Recently, with the rapid advancement in CAD and information science, the field of part libraries exhibits newer and more powerful abilities. Parameterized driving, ontology description, 3D model retrieval, and distributed architecture have been widely applied in this field. As the mainstream method of part library, modeling technology based on parameterized driven is critical to solve the 3D design problems of a large number of standard parts and general parts. Currently, several scholars have built characteristic parts library based on different CAD platforms [17–25]. After several years of 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 support both 2D or 3D files in various formats such as .SAT, .STP, and .iges for thousands of parts.

While at present, the “service characteristics” in the part library have not been well embodied, and there is no unobstructed front-to-back connection among the 3D modeling, transmission, sharing, and usage of parts information. Most service are carried out in the form of usage after downloading from the web, and have not been seamlessly integrated into the design scenarios, which is also one of the main problems to be solved in this paper.

### 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, and standard part library.

The service system of cloud manufacturing general includes five layers, namely, 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 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 and method

Part library here supports the ability of providing various different clients with their needed parts in 2D or 3D formats in corresponding specifications. A prerequisite of this scheme is that the 2D or 3D CAD files can be effectually created according to the practical requests such as modeling parameters and format types. Moreover, the 3D models should be transferred and implanted into 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 are illustrated in Fig. 2.

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

1. Receive the request of part type and corresponding parameter serial from distributed clients all over the web;
2. Create 3D models according to the input parameters in real-time by the ACIS modeler;
3. Transfer the protosomatic ACIS.sat models into the exactly requested formats for practical design scene 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 parameters are nearly infinite. According to the part type and its parameters, the 3D modeling engine will create the 3D models, which can cover all combinations of parameters, avoid the massive modeling and save storage space. Secondly, the clients in proposed framework can be directly embedded into 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 environment on the user terminals is illustrated in Fig. 3, in which the kernel functional module is defined as PLS.

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 this purpose and satisfy the strict requests, we have developed the 3D modeling module, which works without additional support from general CAD software such as NX, Creo, or CATIA to avoid the compulsive pre-requisite

![Fig. 3 The connection between service provider and design environment](image)

| PLS Providers | PLS Hosts |
|---------------|-----------|
| CAA UG Open ObjectARX Pre/ToolKit SolidWorks API | Catia NX AutoCAD Pro/Engineer SolidWorks |
| Part Import | Menu Add-in |
| Intelligent Assembly |  |
| Reverse Edit |  |
| Information Extraction |  |
| 3D Preview | Function Plug-in |
| Model Download |  |
| Order Upload |  |
| Information Extraction |  |
| BOM Service | Embed in |
| eCatalogue Exhibition | Other Applications (such as Word, Excel, or.pdf) |
| Design Assistant |  |
|  |  |
of their installations, and all modules are implemented in the native code or from lightweight open source packages.

PLS provider, the kernel of the framework, is composed of three subsystems: specification data management, 3D modeling kernel, and PLS external interfaces. Specification data management subsystem is more well known as e-catalogue, and 3D modeling kernel is responsible for the generation of 3D part models in various formats according to different design environments. The exported unit of “service” is realized by PLS external interfaces, a package of APIs in format of automation COM, to be invoked by third-party applications on prescribed criterions through the adapters, or connectors that connect PLS provider with hosts, especially with the general CAD platforms such as NX, Creo, AutoCAD, CATIA, and SolidWorks. Moreover, the kernel module can also be wrapped and embedded in the website and cloud scheme, which provides a feasible method for remote designers.

As far as we know, the service scheme on cloud has currently not widely supported in part library as most congeneric solutions are deployed in intrinsic environments, which limit 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 30,000 parts are elaborately organized according to their shapes, usages, material, and other attributes. Multiple modes are provided to screen and access the target part, and the frequently-used 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 has been locked, their modeling parameters can then be input or selected by the designers, and then, the resultant 3D models will be generated in real-time.

With the various 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.

4.1.1 D part modeling and format conversion

Real-time 3D generation and preview of the parts without the support from general CAD systems is a great challenge for part library. In our system framework, 3D part models are first generated with ACIS, one of the most mainstream geometry engines. ACIS holds powerful capability of 3D modeling, supporting entity, wireframe, surface, and other geometric elements. Through flexible handle and operation of these elements, high-precision geometry can be modeling. ACIS employs object-oriented data structure, and C++ is its primary development language; thereby, it contributes an efficient operating and modeling performance. More important, it is an original and lightweight modeler with only dozens of DLLs; therefore, it is very convenient for deployment, especially on the web.

To guarantee that the user-input parameters form a valid geometric combination, a pre-check mechanism has been embedded for 3D modeling. That is, with the users’ continuous input, the mechanism will immediately update the range and accuracy of subsequent parameters to ensure that the input is legal and correct for 3D modeling. Thus, the users will obtain a correct model; however, he inputs or selects parameters as the knowledge on dependency relations and computation rules have been inherent in the modeling modules. 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 then, InterOP(for 3D) and PHLV5(for 2D) are adopted to convert it into various formats including.stp,.iges, dxf, dwg, and hsf. The real-time preview for 3D models is realized based on Hoops with.hsf file.

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. Flexibility in the deployment of modeling module and independence of CAD software are worthy of this cost.

4.2 Geometry and rule constraints

Because the sizes driving the 3D geometric model of parts are interrelated with each other and even contain complex constraints in the format of expression, it is essential to deeply investigate and seek an effective strategy to represent, record, and realize the association relationship. In the
detailed implementation, we employ various kinds of constraints and knowledge to solve the problems.

Primarily, the linking tables for parameter association are the uppermost means to store and manage the geometric relations. The parameters in one 3D model are composed of two categories: active ones and passive ones, and if an active parameter has been modified by the designer, the ranges, values, or even precisions of its associated passive parameters will also be correspondingly changed according to predetermined constraints. A string expression evaluator is embedded in this module to calculate the current value, range, and precision of subsequent parameter. Figure 5 illustrates a parameter table for one typical part, in which its parameters are associated by string expressions.

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 helps avoid repeating 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 constraints of parameter and alternation selection (as shown in Fig. 6) are also recorded in the knowledge database with editing tool (KDT, Knowledge Database Tool) in the format of “IF–THEN” expressions for users. KDT provides active monitoring mechanism for the change of parameters or processing alternations, with which if their values trigger certain rules, the later will automatically come into force, and meanwhile, KDT will take corresponding the per-defined measures to insure the check and correction of parameters.

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 instances, rules, and their knowledge object sets; 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.

Furthermore, the part library also provides a bidirectional channel to exchange information between PLS and its hosts.

---

**Fig. 5** Parameter table for a part

**Fig. 6** Parameter and alternation rules
Feedback edit and successive assembly are two attractive and peculiar operational modes in our implementation. The former provides a mechanism of sending the current design status and information back to PLS provider, and assists it to automatically screen and restore original parameter lists for 3D modeler. Based on the returned information, the designers can realize the parameter edit for this part directly in the exact assembled position. For instance, a component 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 returns back to the interface of parameter list by a pop-up menu; the original assembly information (position, constraint, keep-space, and cut) will be reserved and continue to be applied to the alternate parts. Figure 7 is an example that the old part “LRBS60-10” is replaced by the new one “LRBS100-15” without repeating the assembling process, saving much time.

Data relevance in PLS supports not only parameters editing of one single part, but also “successive assembly.” It is a common design scenario that some components are linked with other ones; the latter are also named “associated parts.” Therefore, if one part has been imported into the design, the likelihood is that its associated parts will also be appended to the product. For instance, once the parameters and specifications of a mold base are determined, the assembling location and constraints of the hexagon bolts with other parts are therefore decided. For these kinds of part series, there is no need to manually import one by one. Aiming at this occasion, successive assembly will help automation to a great degree. When the basic setting of a mold base is initialized, the required parts and their corresponding assembling position are retrieved from associated tables, and afterwards homogeneous matrices are computed, by which the successive parts will be also automatically imported and assembled onto the mold base. The embedded knowledge in successive assembly in an associated table will help to reduce the repeated operations and manual errors, therefore 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, or clients. 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 local environments. Therefore, the PLS adapters bring the two sides together and bridge the gaps between them through 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, and attached documents. 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 or clients can access all the services by importing the APIs into their own frameworks. The architecture of the IoDR is illustrated in Fig. 8.

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. 9 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. For wider adaptability, the concept of “class” has not been applied in the APIs, and the most basic interface form is adopted so that nearly all programming languages or systems can invoke them smoothly.

The part resource deployed in the cloud includes data, programs, and virtual computer devices, constructing a complete run-time environment for part library. These resource and APIs are hosted in an API gateway through gRPC, which aims at providing a unique entrance for remote part library.

![Fig. 8 The architecture of the IoDR](image-url)
and is adopted for routing, composition, and protocol transformation to handle the requests from clients. API gateway also helps to protect the architecture from external attacks and 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 latency and lags 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 technologies of PLS provider and various hosts/clients available, we have implemented a PaaS deploying framework on cloud. The PLS adapters are constructed in accordance with the development toolkits of the hosts as they serve as plug-in programs for the connection between

---

Fig. 9 A package of functions from PLS provider

```
extern _declspec(dllexport) int ShowInterfaceDlg(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 GetParCount();
extern _declspec(dllexport) char* GetParaName(int index);
extern _declspec(dllexport) char* GetParValue(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 GetExecStat();
extern _declspec(dllexport) void ClearData();
extern _declspec(dllexport) void Counters(int index);
extern _declspec(dllexport) void SetRestart();
```
hosts and PLS provider. Through the bridges from PLS adapters, the hosts have the ability of invoking the interfaces exported from the PLS providers, and CAD systems are the frequent 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 10 illustrates typical hosts as AutoCAD, Catia, and NX. The PLS adapter for AutoCAD is implemented with ObjectArx, that for Catia is with CAA 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 models into the ongoing design. By this mean, the remote resources are flexibly linked with local environments.

With the PLS adapter integrated in an actual CAD environment, the generated 3D models of selected part will be directly imported into the ongoing design with various assembling mode, as shown in Fig. 11. A prompt message is illustrated after the importing step for the designers to decide whether its associated parts should be also brought in. If a positive respond returns, they will be automatically imported and assembled.

Since PLS adapters return parameterized 3D models according to users’ selection or input, it is an interesting issue to directly and automatically retrieve information from design scene to reduce manual intervention. Feedback edit and associated assembly in our platform play a partial role, while it is not enough. Mercado et al. [32, 33] have done outstanding work in this field, and they gave us good inspiration. We also developed several rapid design tools based on SolidWorks, NX, and Catia in related projects, and it is an excellent opportunity to integrate them to cooperate with PLS adapters in our later work.

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 sections, 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 a lot of manufacturing enterprises and produces great benefits. This system holds the following characteristics:

1. Cross CAD platform: The 3D modeling does not rely on a specific CAD platform. The part library is separated from CAD modules, and an independent 3D modeling kernel (ACIS) is responsible for the generation of parts. Afterwards, a PLS adapter in different CAD platforms is constructed to import the created model into practical design scenario, avoiding repeat development.
2. Cloud scheme: A identical cloud-based part library service is provided to facilitate the integrated management and update of enterprise data, which is conducive to teamwork. The local deployment has been minimized except the PLS adapter.
3. Coincident operations: With the uniform external interface, PLS can be conveniently and seamlessly implanted in various hosts including most CAD systems without any differences in use. The library is currently 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 also holds some shortcomings that needed further improvements. Firstly, the 3D models are created with ACIS and converted into .stp or .iges, which are not native formats for CAD platforms. Therefore, the imported parts only contain information about geometry and attributes, while their modeling features and structures are missing, which brings inconvenience for edit. Direct 3D driving of native format in CAD software is an interest in our later research. Another, our part library covers only die and mold; currently, more kinds of parts should also be taken into management. As for the further work, mechanical parts of factory automation equipment in manufacturing industry are the main extending targets of our platform.

Author contribution All authors cooperatively developed the software in past several years.

Data availability The data and material has been used in many die and mold enterprises.

Code availability The software application has been used in many die and mold enterprises.

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 for publication All authors consent to publish.

Conflict of interest The authors declare no competing interests.

References

1. Fan Q, Liu G, Wang W (2011) Development of die sets standard parts library based on Pro/E. Procedia Manuf 15:3802–3807
2. Jong W, Wu C, 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, Kim S, Lee K (2012) Integration of evolutionary BOMs for design of ship outfitting equipment. Comput Aided Des 44(3):253–273
4. Ma Y, Tor S, Britton G (2003) The development of a standard component library for plastic injection mould Design Using an Object-Oriented Approach. Int J Adv Manuf Technol 22:611–618
5. Mok H, Kim C, Kim C (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, Zhao X (2013) Cloud manufacturing service platform for small- and medium-sized enterprises. Int J Adv Manuf Technol 65:1261–1272
7. Xu X (2012) From cloud computing to cloud manufacturing. Robot Comput Integr Manuf 28(1):75–86
8. Gao R, Wang L, Teti R, Dornfeld D, Kumara S, Mori M, Helu M (2015) Cloud-enabled prognosis for manufacturing. CIRP Ann Manuf Technol 64(2):749–772
9. Brecher C, Lohse W, Vitr M (2009) Module-based platform for seamless interoperable CAD-CAM-CNC planning. In: Advanced Design and Manufacturing Based on STEP. Springer, pp 439–462
10. Brecher C, Vitr M, Wolf J (2006) Closed-loop CAPP/CAM/CNC process chain based on STEP and STEP-NC inspection tasks. Int J Comput Integ M 19(6):570–580
11. Velde V (2009) Runtime configurable systems for computational fluid dynamics simulations. PhD Thesis. Auckland: Department of Mechanical Engineering, University of Auckland
12. Nasseri A, Newman S, Xu X, Rosso Jr R (2008) Toward interoperable CNC machining. Int J Comput Integ M 21(2):222–230
13. Mokhtar A, Houshmand M (2010) Introducing a road map to implement the universal manufacturing platform using axiomatic design theory. Int J Manuf Res 5(2):252–269
14. Suh S, Shin S, Yoon J, Um J (2008) A new paradigm for product design and manufacturing via ubiquitous computing technology. Int J Comput Integ M 21(5):540–549
15. Lee B, Suh S (2008) 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, Xu X (2012) An interoperable solution for software integration and product data exchange. Enterp Inf Syst 6(3):291–314
17. Han F, Su T, Ou Z, Luo D (2005) Development of standard part library for three-dimensional stamping die by using solidworks. J D Inst L Indust 24(3):211–214
18. Zhang Z, Zhu S, Zheng Q, Dong L (2003) The secondary development of SolidWorks by VC++. J T Inst Techno 19(4):69–71
19. Wang J (2006) Research on feature based information model in part library and its implementation. Dissertation, Harbin Engineering University
20. Lu Y, Liao W, Huang X (2004) Li Y (2004) Research on data description norm and building techniques of web-based parts library. Chin Mech Eng 15(11):987–990
21. Xu X (2006) Research on development and application of distributed library system based on web service. Dissertation, Zhejiang University
22. Zhou J (2005) Research and implementation of part data model library management system. Dissertation, Zhejiang University
23. Cho J, Han S, Kim H (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 Sc 3865:324–333
25. Jin B, Teng H, Wang Y, Qu F (2008) 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. 3D ContentCentral: A free service for locating, configuring, downloading, and requesting 2D and 3D parts and assemblies, 2D blocks, library features, and macros. Dassault Systems. Available: http://www.3dcontentcentral.com/
27. KiM GmbH: A specialist in configurators and 3D visualisation of numerous variants of individual parts as well as subassemblies in 2D and 3D product catalogues. KiM GmbH. Available: http://www.cadclick.de
28. Inpart: A part library providing CAD models of mechanical and electromechanical products and technical documents. Parametric Technology Corporation. Available: http://www.clearpathmedia.com/InPart_Web_Mirror
29. TraceParts: A 3D digital content provider serving engineering, design, procurement, production and maintenance processes and operations. TraceParts Corporation. Available: http://www.traceparts.com/
30. Web2cad: A website providing CAD model of hardware electromechanical products. CADENAS WEB2CAD Inc. Available: http://www.web2cad.co.jp/

31. Protégé: A free, open-source ontology editor and framework for building intelligent systems. Stanford University. Available: https://protege.stanford.edu/

32. Mercado C, Rubio P, Vizan A, Martin D (2017) A new procedure for the automated design of ejection systems in injection molds. Robot Cim-Int Manuf 46:68–85

33. Mercado C, Rubio P, Marquez S, Martin D (2018) A new method for the automated design of cooling systems in injection molds. Comput Aided Design 104:60–86

Publisher’s Note Springer Nature remains neutral with regard to jurisdictional claims in published maps and institutional affiliations.