COMPUTER-AIDED DESIGN & APPLICATIONS
2018, VOL. 15, NO. 6, 872–879
https://doi.org/10.1080/16864360.2018.1462568

Service oriented cross-platform framework for 3D part library based on diversiform knowledge implantation

Wei Liu, Qiang Niu, Chuipin Kong and Xionghui Zhou

National Research Center of Die and Mold CAD, Shanghai Jiao Tong University, China

ABSTRACT

Recent years have seen the great advancement in rapid design and manufacture. As a primary foundation, part libraries play an important role in machine and mold design. Most current libraries are created in specific use environments or CAD platform, which will bring obstacles in the share and exchange of part information. In this paper, we proposed a service oriented cross-platform framework for 3D part library, in which the parts in the library are created in a native ACIS modeler, and then the models are converted into CAD files with various formats, finally these files can be imported into CAD platform for practical design. The whole framework can be divided into three parts, namely, PLS(Part Library Service) provider, PLS adapter and PLS host. Through different adapters, the PLS can be easily grafted on most mainstream applications as a plug-in subsystem or even deployed on the website. In the detailed implementation, diversiform knowledge for part parameter and design is implanted to automatically define the geometry and rule constraints in the parts, with which the users can easily edit or customize the part library according to their needs. The PLS system has been applied in many manufacturing enterprises, and accumulates a large number of successful practical cases.

KEYWORDS

part library service; cross-platform; 3D library; diversiform knowledge

1. Introduction

With the rapidly increasing needs in global product customization and specialization, a limited series of general specifications of industrial merchandises can hardly completely satisfy various customers' requirements. As an indispensable foundation, large number of parts are necessary to be deployed in product assembly. To expediently and agilely access to the parts information, several part libraries or similar systems have been presented, 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 [3, 11, 12, 14, 15].

From the view of detailed implementation, the part library systems 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 period, while on the other hand the libraries inevitably depend heavily on the exact host CAD platform and cannot be transplanted onto another. Therefore, the 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 several isolated island and doesn't hold the ability of communicating with each other. Moreover, information redundancy, mutual contradiction or repeated developments will frequently trouble the designers or enterprises. In addition, they rarely support the idea and mechanism of providing independent their function alone, that is, without any support from CAD platforms or run in the web.

To solve the problems, this paper investigates a service oriented cross-platform (or namely, CAD platform-independent) framework for 3D part library. The term "service" in the context means that its implementation is open to any individual or organization, and is to bridge the gaps amongst users, CAD platforms or other host systems, with which the users can expediently use, add, delete and edit the information for their own application. Moreover, the method to implant design knowledge into the parts is also investigated, with which the users can easily edit or customize the part library according to their 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.

CONTACT Wei Liu liu-wei@sjtu.edu.cn

2. Related work

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. Parameterization, ontology description, 3D model retrieval and distributed/web architecture have been widely applied in this field. Han [4] studied the parametric modeling technology in SolidWorks, and developed a 3D standard parts 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 [16] studied the features and feature models, emphasizing on the feature model of STEP protocol AP214, and analyzed its abstract definition, entity relation and feature recognition in detail. The characteristics in the parts library information model and part information model are also analyzed in detail. Lu [13] 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 model of the parts is given, together with the part feature model and visualization solution based on Web. From the view of system integration, Xu [17] proposed a distributed enterprise database system and a dynamic integration method in the Internet environment. In his paper, a method of service encapsulation and interface construction for parts library system is proposed, on which a part library architecture based on Web service is built. Zhou [19] proposed a parts library system with shared and private mode. The system structure, parts description and data model in library system were studied and set up, based on which the concrete realization is finally given. Cho [1–2] 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 [10] proposed a design reuse approach based on an engineering semantic web and implement a design reuse 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, the ant colony optimization (ACO) 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 years of development, a large number of well-known parts libraries have been established and put into use to the public: 3DContentCentral [5], CADClick [6], Inpart [7], TraceParts [8], Web2CAD [9], etc. They supports 2D or 3D files in various formats such as .SAT, .STP, .iges for thousands of parts.

3. Architecture & methodology

To realize the purpose, we have to construct the applications without any support from CAD system to avoid the compulsive pre-requisite of their installations and all the functions are implemented in our owned way or from a lightweight API packages.

The architecture of the proposed platform is illustrated in Fig. 1, in which the kernel module is PLS. Through the adapters to establish the connections between PLS provider and various kinds of host systems, it can support several kinds of functions such as 3D part modeling and preview, type navigation, BOM, etc.

The PLS provider can be divided into 3 subsystems: type/parameter selection, model generation/ drive and PLS APIs. The first subsystem is more 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 though it is implemented on a non-CAD environment. The exported functions of "service" is realized by PLS, a package of APIs in the 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 with host systems, especially with the general CAD platforms such as NX, Pro/E, AutoCAD, Catia, SolidWorks, etc. In addition, the main functions can also be wrapped and embedded in the website through Asp.net or Ajax.

To our knowledge, the PLS method is currently seldom supported in the field of part library since most congeneric systems are of introverted environment, which limits the applications to a relatively narrow usage occasions.

3.1. Part information management

The system provides various methods to locate the actual part. The most common way to access a part is by a three-level "class-type-parameter" navigator as shown in Fig. 2.

The parts are divided into two main categories, namely mold and die. Fig. 2 is the user interface for mold, as shown there are 35 classes of parts. When accessing one class, we can screen the actual type according to single or combinatorial filters. Whenever the type is selected, the parameters can be input or selected, along with which the corresponding 3D model can be created at the same time.

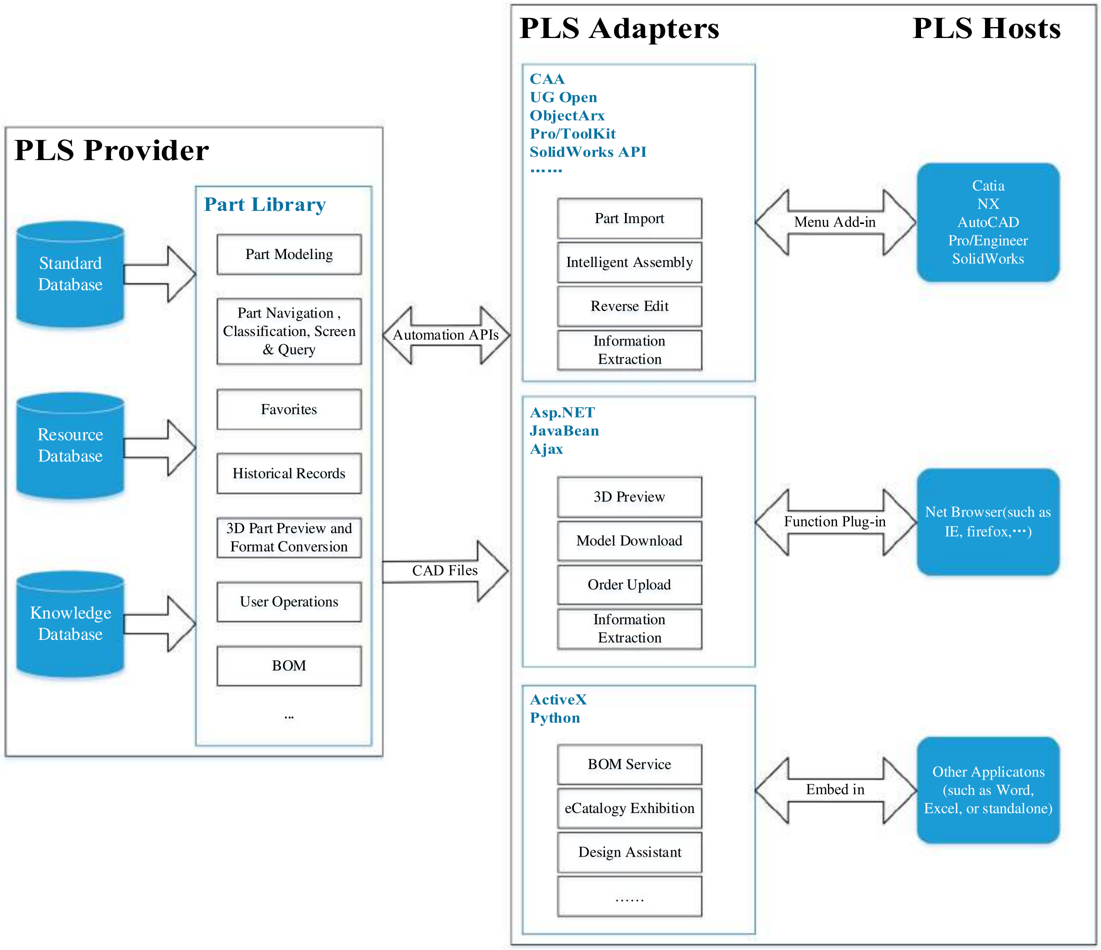

Figure 1. The architecture of the part library service system.

Besides the navigator, users can also access the part through search by type, remark, keywords, etc. The system also records the part selection information, based on which remark, BOM, my favorites and history record can be established.

Remark: a serial of tooltips on the information about the advantages/disadvantages, key points, user specifications which are attached to an individual part.

BOM: the part information about the products. 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 Excel files 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 record: the shortcuts to access the previously visited parts with all parameters reserved in the personal account for each designer.

3.2. 3D part modeling and format conversion

Without the support from built-in CAD systems, the real-time 3D preview and update of the parts is a great challenge. In our implementation, the 3D part models are created with ACIS, a famous geometry engine. ACIS has a powerful ability of 3D modeling, it supports the 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 uses the object oriented data structure and C++ is its development language, thereby, it has 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 parameters list can construct a valid combination, a pre-check mechanism is embedded in the system. That is, although the geometric sizes of a body are in a complicated set of constraints and relations, the

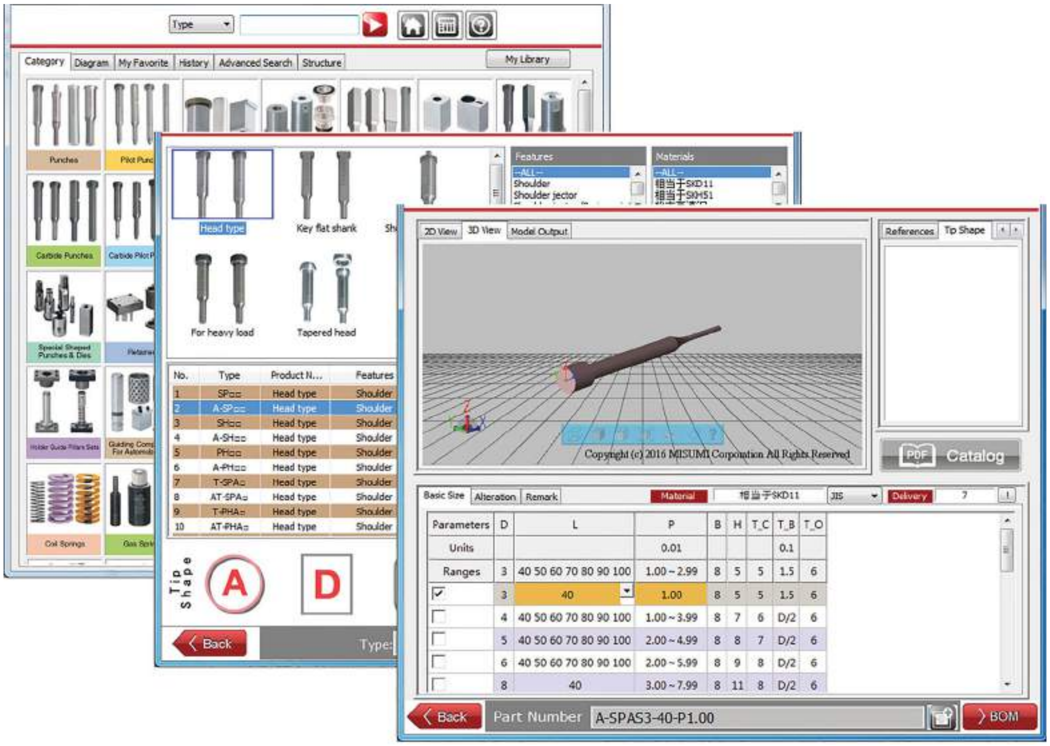

Figure 2. The three-level navigator.

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 3. The formats InterOp R24 supports.

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 systems.

The parameters are input into a modeler, in which .sat file are created, InterOP and RealDWG are then adopted to convert it into various formats including .stp, .iges, dxf, dwg, hsf , etc. The preview of 3D model in the interface is based on Hoops with .hsf file. Fig. 3 lists the formats that InterOP R24 supports.

3.3. Geometry and rule constraints based on diversiform knowledge

As the sizes in geometric shape of any part are interrelated and share many complex constraints, it is necessary to deeply investigate the effective way to represent and realize the relations. In this paper, we adopt various kinds of knowledge to handle the problems.

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 4. Parameter table for a part.

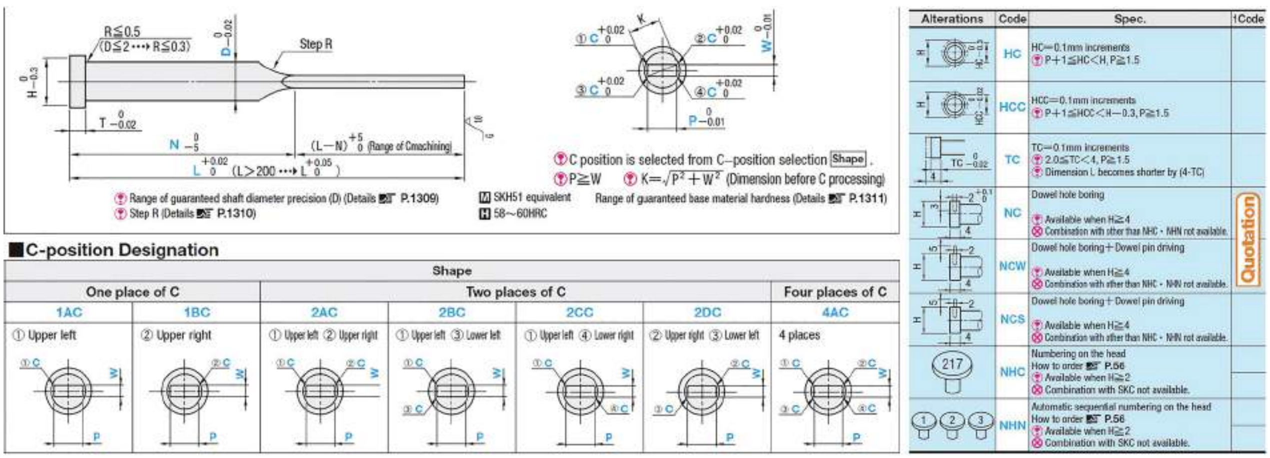

Figure 5. Parameter and alternation rules.

Firstly, the parameter linking tables are the uppermost method to store and handle the geometric relation. The parameters are divided into active and passive ones, and if an active one is change, the range, value, or even precision of its associated passive ones will also be corresponding changes. An expression evaluator is embedded to compute the actual value and precision of each parameter. Fig. 4 illustrates a parameter table for just one part, in which all parameters are associated.

Second, the rules of parameter and alternation selection (as shown in Fig. 5) are also stored in the knowledge database with an editing tool (KDT) for the users in the format of "IF-THEN" expresses. The base parameter setting can also be referenced in Fig. 4. KDT supports an active monitoring for the change of parameters or alternation, with which if their inputs or selections invoke certain rules, the later will take effect and KDT will automatically take corresponding per-defined measures to insure the data correction.

Third, the backward edit and associated assembly will send the current design information back to PLS provider, which also assists PLS to automatically screen suitable parameter lists for 3D modeler. In this cases, the PLS is more like a knowledge receiver and convertor from the real designs. For example, a part has been firstly assembled into the design, if the user wants to change its specifications of size, with the help of PLS provider the only action he need do is just to select the model and change its size on the pop-up interface, the original assembly information (position and constraint), keep-space and cut will act on the new parts. Fig. 6 is an example that the old part "LRBS60-10" is replaced by the new one "LRBS100-15" without the reassembling process.

The knowledge in PLS is not only for the parameters of one part, but for the successive assembly. It is well known

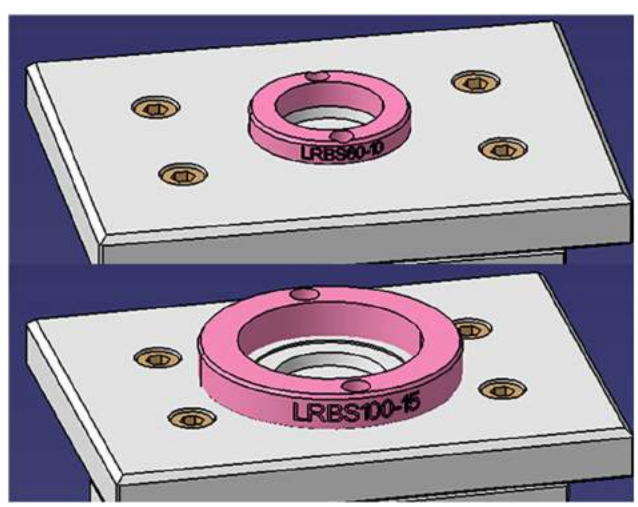

Figure 6. Information reversion in backward edit.

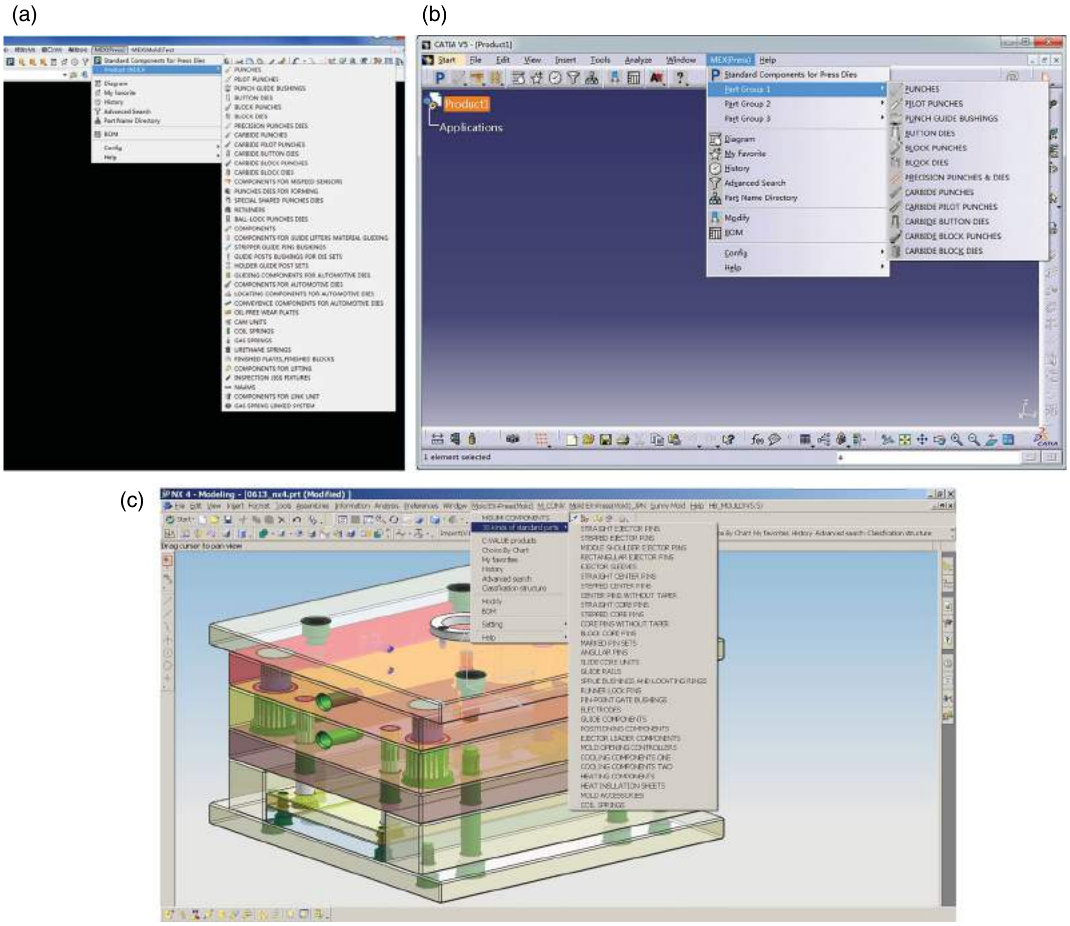

Figure 7. PLS provider hosted in CAD system, (a) AutoCAD, (b) Catia, (c) NX.

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 moldset is decided, the assembling position of the hexagon bolt and other standard parts are also 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 moldset is initialized, the required parts and their corresponding assembling position are queried from the associated tables, and then the homogeneous matrices are computed, by which the parts are automatically imported and assembled onto the moldset.

The knowledge in successive assembly is solidified in the associated tables, which reduces the repeated operations and manual errors, therefore will greatly enhance the effectiveness of large-scale assembly design.

4. Service in CAD platforms

With the kernel functions of PLS, we have implemented three kinds of deploy framework, namely:

Stand-alone version: for the individual user, all the executable programs (including dlls and exe) and data (including tables, figures and pdfs and private date) are deployed on one computer;

LAN version: for a design group, all the executable programs and private date are stored in local computer while the public data (including tables, figures and pdfs) are deployed on the server;

Internet version: lightweight client for the users, all the executable programs (including dlls and exe) and data (including tables, figures and pdfs and private date) are deployed on the cloud, the user access the system through Ajax. The users can handle type selection and parameter setting on the web brower and download the 3D files in various formats.

As for the connection between PLS and application for a realtime application, the PLS adapters are developed according to the hosts' development toolkit as they serve as plug-in programs in the hosts. Through different kinds of PLS adapters, the PLS provider can be deployed in the hosts to invoke the functions exported from PLS provider. CAD systems are the most common users. Fig. 7 is three hosts as AutoCAD, Catia and NX. The adapter

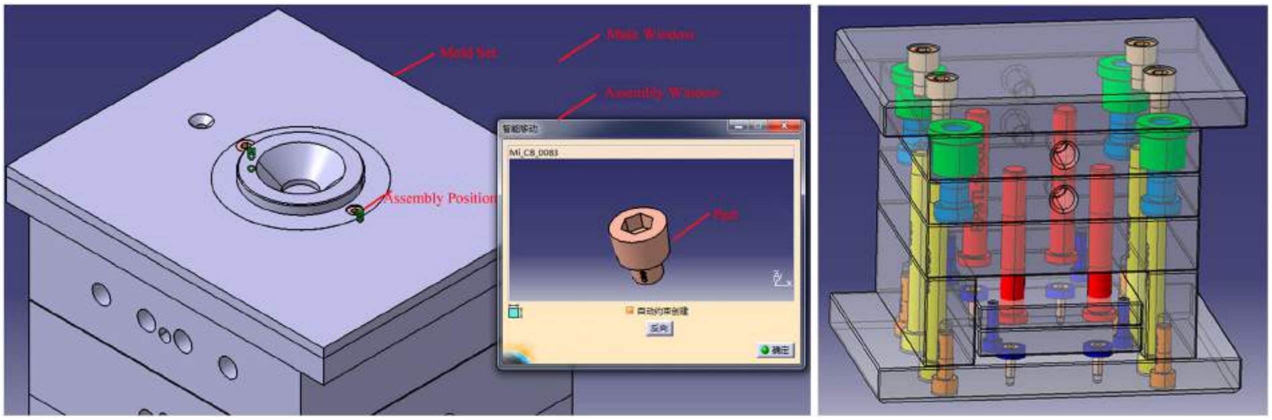

Figure 8. Mold Set Design with PLS system in Catia.

for AutoCAD is developed with ObjectArx, that for Catia is with CAA (Component Application Architecture) and that for NX is with NXOpen.

Integrated in the CAD system, the created 3D part models can be directly inserted into the design with flexible assembly methods, as illustrated in Fig. 8. Whenever the 3D model of a part is created and imported into the design, the prompt message is shown for the user to decide whether its associated parts are needed. If "yes" is responded, they will be automatically assembled.

5. Conclusion

The system has been successfully applied 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 system doesn't depend on a specific CAD platform. The database module and the CAD operating module are separated. An independent CAD modeling kernel is equipped in the system. A rapid expansion and application in different CAD platform is realized to avoid developing several specific systems for different CAD platforms.
(2) Network sharing: A unified network-based database system is established to facilitate a unified 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 system is designed for die and mold design process according to the user's operating habits. Optimization functions are provided for the design process, which greatly enhances the user's design efficiency. Another, the edit work in the CAD systems can be fed back to PLS.

The system also holds some shortcomings needed by improved. Firstly, the 3D models are created with ACIS and converted into .stp or .iges, not the native format, so the imported parts only contain information with geometry and attributes, while its features and structure is missing, which brings the Inconvenience in modification. Direct driving of native model in CAD systems is an interest in our later research. Another, the types in part library cover only mold and die, 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.

ORCID

Wei Liu http://orcid.org/0000-0002-0797-4612

References

[1] Cho, J.; Han, S.; Kim, H.: Meta-ontology for automated information integration of parts libraries, Computer-Aided Design, 38(7), 2006, 713–725, http://doi.org/10.1016/j.cad.2006.03.002
[2] Cho, J.; Kim, H.; Han, S.: Content-Oriented knowledge modeling for automated parts library ontology merging, Lecture Notes in Computer Science, 3865, 2006, 324-333, http://doi.org/10.1007/11686699_33
[3] Fan, Q.-M.; Liu, G.; Wang, W.: Development of Die Sets standard parts library based on Pro/E, Procedia Engineering, 15, 2011, 3802-3807. http://doi.org/10.1016/j.proeng.2011.08.712
[4] Han, F.; Su T.-M.; Ou, Z.-Y.; Luo, D.: Development of standard part library for three-dimensional stamping die by using SolidWorks, Journal of Dalian Institute of Light Industry, 24(3), 2005, 211-214. http://doi.org/10.3969/j.issn.1674-1404.2005.03.016
[5] http://www.3dcontentcentral.com/

[6] http://www.cadclick.de
[7] http://www.clearpathmedia.com/InPartWeb_Mirror
[8] http://www.traceparts.com/
[9] http://www.web2cad.co.jp/
[10] Jin, B.; Teng, H.-F.; Wang, T.-S.; Qu, F.-Z.: Product design reuse with parts libraries and an engineering semantic web for small- and medium-sized manufacturing enterprises, International Journal of Advanced Manufacturing Technology, 38(11), 2008, 1075–1084, http://doi.org/10.1007/s00170-007-1157-y
[11] Jong, W.-R.; Wu, C.-H.; Li, M,-Y.: Web-based navigating system for conceptual mould design with knowledge management, International Journal of Production Research, 49(2), 2011, 553-567. http://doi.org/10.1080/00207540903439830
[12] Lee, J.-H.; Kim, S.-H.; Lee, K.: Integration of evolutional BOMs for design of ship outfitting equipment, Computer-Aided Design, 44(3), 2012, 253-273. http://doi.org/10.1016/j.cad.2011.07.009
[13] Lu, Y.; Liao, W.-H.; Huang, X.; Li, Y.-G.: Research on data description norm and building techniques of Web-based parts library, China Mechanical Engineering, 15(11), 2004, 987-990, http://doi.org/10.3321/j.issn:1004-132X.2004.11.013
[14] Ma, Y.-S.; Tor, S.-B.; Britton, G.-A.: 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), 2003, 611-618. http://doi.org/10.1007/s00170-003-1555-8
[15] Mok, H.-S.; Kim, C.-H.; Kim, C.-B.: Automation of mold designs with the reuse of standard parts, Expert Systems with Applications, 38(10), 2011, 12537-12547. http://doi.org/10.1016/j.eswa.2011.04.040
[16] Wang, J.-D.: Research on feature based information model in part library and its implementation, Dissertation, Harbin Engineering University, 2006, http://doi.org/10.7666/d.y936925
[17] Xu, X.-R.: Research on development and application of distributed library system based on Web service, Dissertation, Zhejiang University, 2006
[18] Zhang, Z.-P.; Zhu, S.-H.; Zheng, Q.-C; Dong, L.-M.: The secondary development of SolidWorks by VC++, Journal of Tianjin Institute of Technology, 19(4), 2003, 69-71. http://doi.org/10.3969/j.issn.1673-095X.2003.04.020
[19] Zhou, J.-H.: Research and implementation of management system for part data library, Dissertation, Zhejiang University, 2005