Since January 2020 Elsevier has created a COVID-19 resource centre with free information in English and Mandarin on the novel coronavirus COVID-19. The COVID-19 resource centre is hosted on Elsevier Connect, the company's public news and information website.

Elsevier hereby grants permission to make all its COVID-19-related research that is available on the COVID-19 resource centre - including this research content - immediately available in PubMed Central and other publicly funded repositories, such as the WHO COVID database with rights for unrestricted research re-use and analyses in any form or by any means with acknowledgement of the original source. These permissions are granted for free by Elsevier for as long as the COVID-19 resource centre remains active.
Abstract

Since the Ministry of Education launched Education Informatization 2.0, the digitalization of colleges and universities has entered a stage of rapid growth. However, after more than 20 years of construction, problems such as system barriers and information islands have emerged in the digital construction of university systems. In order to solve such problems between the university systems, this paper proposes an easily expandable and configurable open information integration architecture by considering traditional information integration methods and combining with Web service technology. The architecture handles user service invocation information through a service layer, and manages the registration and invocation of services through a service module. The permission module manages user permissions to prevent information leakage and security issues. The data module abstracts data-related services to provide a basis for the deep use of data. And other optional development services are designed to satisfy special requirements for different platforms. The architecture proposed in this paper can integrate different heterogeneous subsystems in colleges and universities, eliminating the problem of system barriers and information islands, and providing specifications for the construction of new applications.

© 2020 The Authors. Published by Elsevier B.V.
This is an open access article under the CC BY-NC-ND license (https://creativecommons.org/licenses/by-nc-nd/4.0)
Peer-review under responsibility of the scientific committee of the KES International.

Keywords: Application integration, Education Informatization, Information platform, Web service;
1. Introduction

Since the human society entered the 21st century, information technology has penetrated into all aspects of people's lives. Information technology has played a profound role in both of our daily or production life. Affected by the impact of information technology, the education industry also has to take the pace of digital construction. Building a smart campus has become an essential method for the development of college education.

After nearly 20 years of smart campus construction, the digital construction level of Chinese colleges and universities has been significantly improved. However, the digital construction of colleges and universities encountered many problems[1], which fundamentally restricted the development of digital construction in colleges and universities:

1. The problem of data redundancy and data islands caused by the disconnection of subsystems in colleges and universities;
2. The problem of low user experience and work efficiency caused by the complicated subsystem entrances.

These problems not only appear in the process of digitization in colleges and universities, but also in the process of digitization in many companies and enterprises. Therefore, Enterprise Application Integration (EAI) came into being. EAI is a method and technology that integrates heterogeneous applications which based on different environments and different programming languages[2]. Companies and enterprises use application integration technology to solve the problems caused by the isolation of multiple heterogeneous subsystems, avoiding the huge cost of reconstructing all systems together, as it is an more efficient and cheap solution. Facing similar problems in the education industry, colleges and universities also urgently needs an efficient solution. Therefore, the application of application integration to the digital construction of colleges and universities is the best way to coordinate the two aspects based on the actual situation and solve the above problems.

However, the traditional application integration solution still has some problems when it is applied to the education industry[3]: for example, when a new system needs to be integrated, the traditional application integration solution requires a series of complex configurations, but the university does not have professional personnel who focus on maintaining the integration platform; most of the system data of colleges and universities are related to the personal privacy of students. But the traditional application integration schemes lack permission management, which will lead to the leakage of student privacy; the traditional application integration schemes are based on information interaction between systems, often without considering the deep mining of data in the era of data science, lacking of support for data analysis[4].

After fully understanding the relationship between applications in the education industry, as well as extensively investigating the advantages and disadvantages of various integration methods, this paper designs and builds an open and extensible information integration platform architecture that can integrate existing Heterogeneous systems of different services and various applications which are integrated in order to break the barriers between systems and systems, data and data, eliminate the problem of information islands, give full play to the functions and advantages of existing systems, and standardize new systems and new applications Construction models and standards.

The main contributions of this article are:
1. Proposed a set of loosely coupled, application extension simple application integration architecture for the education industry, which solves the problems of system barrier and information islands among college systems;
2. The permission module in the integrated platform is proposed to provide guarantee for data security;
3. The data module in the integrated platform is proposed to provide support for data analysis.

2. Architecture construction

In order to give full play to the functions of the existing heterogeneous systems of different businesses in colleges and universities, how to realize the communication between these heterogeneous systems and the data exchange of these systems without affecting the normal management functions and teaching order, is the main research issues of this article.

The traditional application integration methods can be divided into three types: information-oriented integration, process-oriented integration and service-oriented integration.
Although these three types of application integration methods can connect existing systems between enterprises and eliminate the problem of information islands to a certain extent, they still have their limitations[5], such as:

1. The construction of information integration is too complicated, and it will affect the normal work of the original systems, resulting in chaos in the management order and teaching order of the university;

2. For newly-built systems or applications, the integrated platform needs to be rebuilt, and the scalability is poor, and universities lack personnel for continuous maintenance of the platform;

3. No consideration is given to permission issues, which may easily lead to information leakage and cause hidden dangers to the privacy of students and staff.

Therefore, based on the thinking of the traditional information integration model, combined with the popular Web service technology, this paper proposes the information integration platform architecture of Figure 1 applied to the education industry.

![Fig. 1. Architecture of integration platform.](image)

The architecture consists of the platform front-end, a service layer, three sub-modules and other services. The service layer is used to handle users’ web services; as for the three sub-modules, the permission module is used to manage permission such as checking permission and authorization, the service module is used to manage services such as service registration and search, and the data module is used to provide services such as data search and data processing; Other services include mail services, data transmission, notification services, and service customization.

This architecture is easier to expand and maintain than the traditional application integration architecture, and the permission module ensures the security of system data, and other services can be customized to make the information platform more diversified. Through the realization of this architecture, this paper lays a solid foundation for the realization of the information integration platform of colleges and universities and even the deep use of data in the future.

3. Modules

This chapter will disassemble the proposed system integration architecture of colleges and universities, introduce and implement the structure of each part.

3.1. Platform frontend

The front-end of the platform is a display interface that provides services info. After the user logs in, the back end will return to the user to view the content according to the permissions. The user can find the service he wants to invoke on the front-end, and then click the invoke button to send the API to the back end for permission verification, and then the user can choose the processing method for returning the result. On the front-end, users can register and
authorize services on the visual interface. Registration requires the service registrar to upload the WSDL file, and authorization can authorize the registered service to others.

For other services, users can schedule through the front-end, for example, send emails regularly, add the service information they need in the task customization bar to allow the corresponding department to develop, transfer the data they need, and so on.

In addition, the front-end can be used to include the front-end pages of other systems, using the idea of micro front-end to integrate the websites of other systems[6], not only does not need to significantly modify the front-end of other systems, but also provides a unified website entrance to avoid complicated website system.

3.2. Service layer

The service layer is a normal web backend, used to receive APIs sent by the front-end. After the front-end API is sent to the service layer, the service layer first queries the service module for specific information about the service. The information includes: service name, service usage, service address, service parameters, service provider, etc. After obtaining the specific information of the service, the service layer will confirm the permission to the permission module. If the user has the permission to make the service invocation, the service layer will combine the service information and then make the service invocation to the address provided by the service provider (This process can decide whether the process is synchronous or asynchronous according to the registration information of the service), and then return the result to the user.

In addition, the service layer can be configured on multiple servers to cope with a large number of users simultaneously. You can also use components such as Nginx for load balancing and k8s for container scheduling to further increase the load on the service layer.

Compared with the traditional information integration methods, the integration scheme using this Web Service[7] can realize a loosely coupled application integration framework through the service definition, publishing, discovery, binding, invocation and other mechanisms. This implementation is easier to expand than traditional application integration solutions, and it proposes standards and specifications for the construction of new application systems.

3.3. Service module

Service module mainly manages service issues. In general, the main operations of services are divided into service registration and service invocation. Due to the different operators and processes involved in service registration and service invocation, this section separates these two operations.

Service registration:

The structure of service registration is shown in Figure 2:

The service registration process is as follows:

First, the service supplier (usually the development engineer of each system) develops the service function. The service must be accessible using Web Service. After the development is completed, the service supplier generates a WSDL description document (usually in XML format) based on the service developed by itself, and then hands the registration file to the service registrar (usually the administrator of each system). The service registrar uploads the WSDL document[8] to the service layer through the front-end entrance of the information platform, and the service layer uses a customized XML parsing module to parse. After the analysis is completed, the service layer will generate a corresponding piece of service information based on the registration information and service registrar information, and send it to the service warehouse for storage. The service warehouse is usually a UDDI registration warehouse[9]. Information integration platform developers can decide whether to use public UDDI or private UDDI or re-implement a private UDDI registration center with more customized functions according to their needs.
For services that already exist in each system and are not accessed by Web Service, the service provider needs to repackage the service or redesign the service access method.

![Service registration construction.](image1)

**Service registration construction.**

![Service invocation construction.](image2)

**Service invocation construction.**

### Service Invocation:

The structure of service invocation is shown in Figure 3:

The process of service invocation is as follows:

1. Users and other systems (a channel can also be designed specifically for the system level) first obtain service information through the login information platform, and then select the required service to invoke. The service information and login information will be sent to the service layer through the API. The service layer obtains the information of the service invocation, and then search the detail information of the service to in service warehouse.

2. If the service information does not exist in the service warehouse, the service layer will directly return the result of the failure result to the user (this situation will almost never happen because the service displayed by the front-end is filtered unless the system has an exception). If the invoked service exists in the service warehouse, the service layer will obtain the service information returned by the service warehouse, and confirm to the permission module whether the current user has the authority to invoke the service. If the user does not have the permission of the service, the service layer will return the call failure information to the user (this situation usually does not happen, the reason is the same as above); if the user has the permission of the service, then the service layer will parse the service information returned by the service module to an specific service invocation. Then service layer will sent the invication to the service supplier, and the result is returned to the service layer in a synchronous or asynchronous manner. The service layer then returns the result of the service invocation to the user.

In the whole process, the user will only send one API for service invocation, and other operations are handled by the information integration platform, which makes the user do not need to care about which subsystem the logic for implementing the service is under, as if all systems are integrated together. This is the benefit of using Web Service to implement an integrated platform[10].

### 3.4. Permission Module

Traditional enterprise application integration methods often lack permissions management[11], because the traditional application integration method often uses a lot of middleware, channels, and database connections. This kind of hard integration method can’t improve the logic function of the systems. Such integration will cause the use of services and data unclear, which may easily lead to the risk of data leakage[12]. Even all systems will be exposed to danger due to all systems integrated together. Therefore, this article designs a permission module based on the way of Web Service to protect the system security and avoid information leakage.

First of all, each user will have an `<Account>`, this `<Account>` records the user's username, password and information of the department. At the same time, each user will have their own `<Role>` information, for example, the ordinary user’s `<Role>` can be a student or teacher, the management teacher’s `<Role>` can be an administrator,
and the data user's <Role> can be data analyst. Since there may be a situation where an account has multiple <Role>s, the relationship between <Account> and <Role> is shown in Fig. 4.

Fig. 4. Relation between Department, Account and Role.

<Permission> is the core of the permission module. As shown in Fig. 5, <Permission> represents whether a <Role> can invoke certain APIs. When a <Role> does not have the permission of an <API>, there is no foreign key association between the <Role> table and the <API> table in the <Permission> table. On the contrary, if there is the permission of the <API>, the <Permission> table will have the foreign key association record of this <Role> and this <API>. When the service layer receives the service invocation request and then finds the service information through the service module, the service layer will request permission check to the permission module. Only when the permission exists, the service can be invoked normally.

Fig. 5. Relation between Role and API (Permission).

In the information platform framework, service invocation, service authorization, and service registration all require permission verification, and permission settings are based on <Role> and <API>. In simple terms, one only need to authorize the provided <Service> to a certain <Role>, then all <Account> belonging to this <Role> will have permission to call this <Service>. In other words, the service is bound to the role, and the service is bound to the identity. Only when this <Role> is obtained can there be a corresponding <Permission> to complete the service that only this <Role> can call, so that the <Account> The practice of directly binding with <Service> avoids the trouble of repeated authorization and complicated table structure. At the same time, <Role> and <Service> are many-to-many relationships. One <Role> can correspond to multiple <Service>, that is, one role can use multiple services, and one <Service> can correspond to multiple <Role>, namely The same service can be called by different roles[13].

The design of the permission module is shown in Figure 6, which is mainly used to verify the permissions of user actions, so as to avoid the risk of information leakage to a certain extent. And through the filtering function of the permission module, the user can be easily aware of his identity, and it is easier to know what services he/she can invoke, which improves the user experience to a certain extent.
Of course, the function of the permission module can’t be limited to this. Other functions designed in the framework can also use the permission verification of the permission module to limit the functions. For example, the administrator has the permission to send emails. The data analyst has the permission to dynamically configure the database and cloud services, service invoker can apply to service suppliers for the development of new services, etc.

3.5. Data module

The data module can be regarded as an architectural approach that combines the data-oriented integration model and the Web Service technology in traditional application integration. Based on the service module, this module encapsulates the data-related services independently, so that the data-related services are logically separated from other services, and other data-related tasks are added, such as dynamic configuration, database and configurable cloud services. Drawing them into a module independently helps the operator to understand the data usage on the one hand, on the other hand it can also make the data usage more diverse and the depth of use more optimized, so as to make full use of the characteristics of the data[14]. In recent years, the concept of "data concourse" led by Alibaba has accelerated the wave of data integration among various enterprises. The core idea of "data concourse" is de-specialization of data, universal use of data, and deep use of data. The construction of the data module in this article is based on the Web Service technology, making the invocation of the information platform data without the restrictions of the subsystem architecture, language, and database type, which fits the idea of "data concourse". The idea of configurable database allows all kinds of data to be cleaned and put into the information platform database, which allows data analysts not to frequently use service invocation operations. Among other services, information platform developers can selectively configure high-performance servers according to the needs of data analysts and their own resources to provide data analysts with a completely online data analysis platform.

The main structure and operation of the data module are shown in Figure 7.

Users invoke data-related services through the front-end of the information integration platform, and the invokes are sent to the service layer through the API. The service layer receives the invoke sent by the user and the user's login information, and then queries the service module for detail service information. The service module receives the query request from the service layer and returns detailed information about the related service to the service layer. The service layer uses the detailed information of the service to confirm the permission to the permission module. After the confirmation is passed, it will use the detailed information of the service to parse into the API invoked by the information and send it to the service supplier. The service supplier executes the information invoke after receiving the information call API, and then sends the data required by the user back to the service layer. When the data is returned to the service layer, the user can process the data according to the needs, can download the data directly (when the data is relatively small), or can save the data directly to the database of the information platform.
(can be divided into relational types according to the configuration Database and non-relational database need to be configured in advance), or save the data to the cloud (also need to be configured in advance).

In some cases, for data that changes infrequently, such as information about students who have graduated, information about the residence of faculty and staff, or information about the number of teachers and students in the school, if the data analyst uses these data, it needs to be repeated every time. It is obviously unnecessary and time-consuming to invoke services every time on these data. Therefore, we can build a layer of cache database for these data, and when the data analysts need these data, directly return the content to the cache layer; in order to avoid data inconsistency, you can set the expiration time of the data, every once in a while Refresh the data in the cache layer.

All in all, the data module is a module built specifically for data integration, which facilitates the exchange of data between various systems in many ways. For some functions that can be added, the developers who need the information platform are based on practice and needs, and develop based on actual needs.

3.6. Other services

Other services are a module that mainly serves platform management and data analysis, as well as a module that can be developed according to real needs. The construction of application integration platforms should be based on practice. Starting from the project, the additional functions of application integration are decided according to the real situation of the project, especially when applied to the education industry. The module of other services gives platform developers enough freedom to implement the specific functions required by the project. In fact, developers can conduct selective development or implement what they need. Listed below are some module functions.

**Mail service:**
Mail service is a function that many systems have. Administrators and teachers can send the information of platform maintenance, update and special circumstances to all students and faculty through the mail service. This is more effective than an announcement. Mail service can be implemented using Java Mail with mail service providers (such as MailGun, Amazon SES and SendCloud etc.);

**Notification service:**
The notification service can be scheduled and customized to remind users. For some time-consuming asynchronous services, users do not need to stay on the current page, but can notify the user of the final result of the asynchronous service through the notification service[15]. Implementation can use Quartz[16] and other task scheduling components, and prompt with mail service;

**Data transfer:**
After configuring the information platform database, the data is transferred between the server-server and server-local through the visual interface[17];

**Service customization:**
Data analysts can publish customized service messages according to their own needs[18], and relevant departments will develop and register and authorize services after seeing them. Service customization can not only untie the service from the information platform, letting the department focus on its own service development, but also allow the data analyst to focus on its own data analysis work, avoiding spending too much time in the communication of service development.

4. Related work

With the continuous deepening of the enterprise's digitalization process, the enterprise encountered problems such as system isolation and information silos, so EAI was born. In 2001, Johannesson and Perjons put forward the design principles of EAI[19], laying the main development direction of EAI. In 2003, Irani, Z proposed the important impact of EAI on the life cycle of information systems[20], and EAI has gradually become the main method for enterprises to solve problems. Gradually, people have proposed three main ways of traditional application integration[2], which has become the industry's three main axes of application integration. Later, with the gradual development of Web service, people began to gradually associate EAI with Web Service. He, Wu and others[21] compared a variety of application integration solutions and found that Web Service technology has great benefits for EAI implementation. Now, enterprise application integration solutions based on Web Service have become the mainstream of society.

5. Conclusion

In order to solve the problems of system isolation and information islands in the university system in the education industry, this paper proposes an open and easily scalable application integration architecture. The architecture is mainly based on Web Service technology to achieve loose coupling and high compatibility between subsystems in colleges and universities. The service module is designed to make the registration and invocation of services more streamlined, and to make the maintenance of services easier. A permission module is designed to authorize the use of service calls and other functions, which ensures the security of the information of teachers and students in colleges and universities, and also makes users more aware of their permissions. The data module is designed, and services and facilities focused on data are extracted, which greatly facilitates the daily work of data analysts and promotes the in-depth use of data. The mutual cooperation of modules and the supplementary support of other services have perfectly solved the problems that need to be solved urgently in the digital construction of colleges and universities.

In the future, we will gradually apply this architecture to the construction of the information platform of East China Normal University. In practice, we will continue to find problems, solve problems, continuously optimize and upgrade the architecture, and design a general application integration architecture for the education industry.

Acknowledgements

This work is partly supported by the National Natural Science Foundation of China (61502236, 61672234, 61877018, 61977025, U1811264), the National Key Research and Development Program of China (2016YFB1000905), Shanghai Agriculture Applied Technology Development Program, China (T20170303). The authors would like to thank the classmates and teacher at the Lab and the people who fighting the COVID-19 for their help.

References

[1] H.D. Xiong, H. Mo. (2011) “Current Situation and Development Trend of Construction of University Informatization.” Journal.
[2] Y. Gu, S. Zhang, X. Zhu. (2003) “A Study of Enterprise Application integration(EAI) Solution.” Computer Engineering and Application 39(6): 209.
[3] Z.G. Chen, H.P. Cheng. (2009) “Research on Enterprise Application Integration Categories and Strategies.” Journal.
[4] M. Zhang, P.J. Chen, Research on the Informatization Construction of University Archives Management, in: X. Xiao (Ed.), Proceedings of the International Conference on Education Innovation and Social Science, Atlantis Press, Paris, 2017, pp. 342-345.
[5] P. Wuliang, Z. Li, W. Lei. (2007) “Survey on enterprise application integration technology.” Application Research of Computers 24(9): 12-15.
[6] Y. Caifang, L. Chuanchang, S. Zhiyuan. (2019) “Research and Application of Micro Frontends.” IOP Conf. Ser., Mater. Sci. Eng. (UK) 490: 062082 (6 pp.).
[7] K. Yue, X.-L. Wang, A.-Y. Zhou. (2004) “Underlying techniques for Web services: a survey.” J. Softw. (China) 15(3): 428-442.
[8] S. Kanimozhhi, A. Kannana, K. Selvamani, A. V. Kumar. (2015) A NOVEL APPROAVIH TO DISCOVER WEB SERVICES USING WSDL AND UDDI, in: S. Patnaik (Ed.), International Conference on Computer, Communication and Convergence, Elsevier Science Bv, Amsterdam, pp. 480-488.
[9] K. Sun, D. Chen. (2003) “Application Model Research Based on UDDI and Web Service.” Application Research of Computers 20(5): 133.
[10] W.U. Jun, D. Chao, S. Xinyu, Y.O.U. Benshan. (2006) “Research on Enterprise Application Integration Based on Web Services.” Application Research of Computers 23(8): 64-66.
[11] J.M. Seog, K.Y. Deok. (2013) “Design and Implementation of EAI(Enterprise Application Integration) System for Privacy Information.” KIPS Transactions on Computer and Communication Systems 2(1): 51-58.
[12] F. Yongze. (2006) “Research and Realization of Privilege Management in PDM System.” Journal of Engineering Graphics 27(3): 48-54.
[13] G. Liu, R.N. Zhang, H.M. Song, C. Wang, J.H. Liu, A.J. Liu. (2016) “Ts-RBAC: A RBAC model with transformation.” Comput. Secur. 60: 52-61.
[14] J. Han, M. Kamber, J. Pei, Data Mining: Concepts and Techniques, 3rd Edition, Data Mining: Concepts and Techniques, 3rd Edition, Morgan Kaufmann Pub Inc, San Francisco, 2012, pp. 1-703.
[15] C.-y. Liu, H.-j. Wang, J.-x. Chen, X.-j. Zou. (2010) “Workflow and Service Oriented E-mail System Model.” Comput. Eng. (China) 36(15): 68-71.
[16] Z. Wang, Y. Zhu, J. Song, W. Liu. (2008) “Optimal method on dynamic maintenance task scheduling.” Chin. J. Mech. Eng. (China) 44(1): 92-97.
[17] X.D. Qin, Y.Y. Luo, N. Tang, G.L. Li. (2020) “Making data visualization more efficient and effective: a survey.” Vldb J. 29(1): 93-117.
[18] L. Haiqi, S. Wei, Z. Xin, J. Zhongbo. (2006) “A policy framework for collaborative Web service customization.” 2006 2nd IEEE International Symposium on Service-oriented System Engineering: 8 pp.-8 pp.
[19] P. Johannesson, E. Perjons. (2001) “Design principles for process modelling in enterprise application integration.” Inf. Syst. 26(3): 165-184.
[20] Z. Irani, M. Themistocleous, P.E.D. Love. (2003) “The impact of enterprise application integration on information system lifecycles.” Inf. Manage. 41(2): 177-187.
[21] W. He, L.D. Xu. (2014) “Integration of Distributed Enterprise Applications: A Survey.” IEEE Trans. Ind. Inform. 10(1): 35-42.