Available online at www.sciencedirect.com

ScienceDirect

Procedia Computer Science 176 (2020) 1567–1576

Procedia Computer Science

www.elsevier.com/locate/procedia

24th International Conference on Knowledge-Based and Intelligent Information & Engineering Systems

A Novel Application of Educational Management Information System based on Micro Frontends

Daojiang Wang[a], DongMing Yang[a], Huan Zhou[a], Ye Wang[a], Daocheng Hong[a,b], Qiwen Dong[a], Shubing Song[c,*]

[a]*School of Data Science and Engineering, East China Normal University, Shanghai 200062, China*
[b]*Information Technology Services, East China Normal University, Shanghai 200062, China*
[c]*Graduate School, East China Normal University, Shanghai 200062, China*

Abstract

With the launch of the Education Informatization 2.0 action plan by the Ministry of Education, a large number of college information systems have been born in China. Most of these systems are single page web applications (SPA) based on traditional MVC structures. Due to the complex logic and high coupling between educational businesses, developers need to write a lot of code. The education information system has many businesses and high coupling between businesses that the system often face problems such as bloated frontend businesses, iterative system updates, and difficult incremental function developments. Combined with the idea of service-oriented architecture, this paper proposes a micro frontends solution and applies it to the new generation of graduate information platform of East China Normal University, which has better agile development capabilities. From the aspects of service separation, efficient development, and incremental upgrade, this paper verifies that the architecture can well adapt to the needs of future educational management information system. The design of the micro frontends provides a new idea for the development of a new generation of education information system.

Keywords: Educational Management information system, Micro Frontends, Education Informatization, Graduate information system;

* Corresponding author. Tel.: +86-18018842137.
E-mail address: shbsong@cs.ecnu.edu.cn

10.1016/j.procs.2020.09.168

1. Introduction

In the era of "Internet+", the education informatization is an important part of internet construction [1][2], and Chinese education information system has made significant progress in research and development. Major universities and colleges have successively built education administration systems, scientific research systems, office systems, and school platforms [3][4], see Fig. 1. Through the application of information technology, the office efficiency, management efficiency and service levels of universities have been greatly improved [5]. However, in the current construction of information education, most education information systems only focus on a single topic and simple tasks. The use of the technical architecture is at a standstill, and high-availability web-server clusters and DB clusters are often added for system performance improvement. The system constructions are centralized, high-cost, repeating the function integration and data integration. These works are very easy to cause repeated construction and information isolated island. At present, Chinese education institutions are facing the expansion of the enrollment scale, the updating of functions [6]. And the university business processes are also undergoing in-depth standardization and reform, the information system needs to be continuously implemented and improved. These requirements require that the education information system must be capable of iterative upgrade to avoid obstructing reforms [7]. In the solution of this paper, we use the idea of micro frontends to decouple applications from the system, and independently run and deploy the functional applications of the system, so that the system will have higher scalability and better performance [8].

Under this background, East China Normal University has developed a new graduate information system and explored new practices of information technology in the educational management information system [9]. In order to build an efficient and highly scalable information system, we will study the application of the micro frontends architecture in graduate information systems, and find corresponding solutions to the frontend technical problems of the educational management information system in this paper. The main problems faced by the existing educational management information system are: 1) the information system has complex business logic [10]. For example, the business of the graduate information system involves the whole period information of the students at school, which will generate a complex business chain. The load business system put forward higher requirements for frontend services; 2) the system needs to build many functional systems to meet all aspects of intelligent informatization in universities and colleges [11]. Fig. 2 shows our university's basic information system; 3) with the advancement of education informatization, the system will generate new functional requirements, which cause the entire system to frequently add new functions, and release. It is difficult to perform agile development, and it will bring greater challenges to the infrastructure. In order to solve the above problems, this paper proposes an educational management information system based on a micro frontends architecture, divides the application based on the bounded context, develops the system as components, and applies it to a new graduate information system. The practice has proved that the micro frontends architecture can solve the problems of backward business functions, high iteration costs, and low scalability of the original graduate information system of East China Normal University. And the new graduate information system meets the business needs, supports automated continuous deployment, personalized services, and rapid iterative upgrades. The main contributions of this paper include the following three points:

- Propose the development idea of educational management information system based on micro frontends, which solves the problems of high coupling, difficult delivery and poor scalability;
- Propose the method of business separation and component development based on the bounded context realizes the efficient development and iterative upgrade of the system;
- Implementation of the graduate information system of the East China Normal University based on the micro frontends.

Fig. 1. Various and diverse websites in education information system

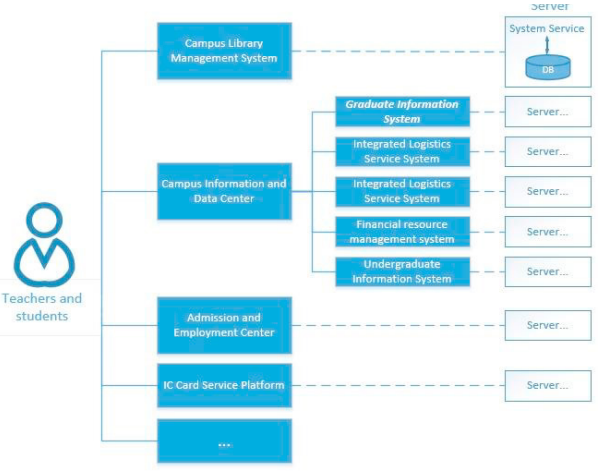

Fig. 2. Education information system

2. Micro frontends

In recent years, microservice have gradually become a trend in software architecture design. Microservice are generally considered as a collection of independent services, and each service is an independent process that focuses on a single responsibility and functionality, decoupling the dependencies between services. Microservice have unique advantages for medium and large applications. This idea of splitting has shifted to the frontend, but requires different understanding and application [12]. In information systems, backend microservice use component design to decouple the dependencies between services, but in contrast, frontend microservice require aggregation. Nowadays, the system often provides a series of applications for different objects, services, and functions. For user convenience and ease of use, websites need to aggregate applications on the same page, which is also a disadvantage of traditional application frontends. Traditional frontends often use a single web application (SPA) to implement a feature-rich system [13]. Although this design can provide a good user experience, it often blurs the functional boundaries and business redundancy. In the later stage, the frontend layer is more difficult to maintain and has poor scalability. Each change requires the entire project to be re-deployed.

Frontend research is an evolving field. Component-based pages are considered popular designs today. Some important web development frameworks (Angular, React, Vue, etc.) use component-based methods as a reliable way to develop web applications [14]. Today, the microservices backend has a more mature solution. Combined with the current development model of front-back stage decouple, the micro frontends is a good idea and technology [15]. For the many problems exposed by the traditional education information system, such as technical changes, component upgrades, development team replacements, and the need to add new functions and applications, the micro frontends can provide a good solution with incremental upgrade, independent development and deployment. When developers use the micro frontends architecture to develop independent component pages and deploy application modules, they only need to focus on the relevant functions and components, so that the system can be quickly released. The design concept of the micro frontends is embodied as Be Technology Agnostic and Isolate Team Code. The developers can independently choose a technology stack for

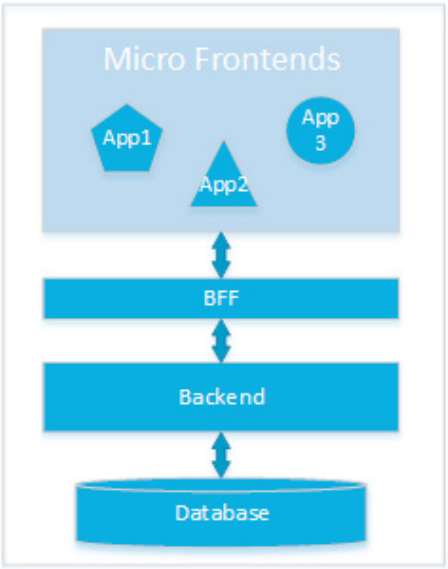

Fig.3. Micro frontends architecture

independent development, build independent programs, establish named isolation, and use browser event communication. Fig. 3 shows the main design of the current micro frontends.

Micro frontends are slicing up big and scary things into smaller, more manageable pieces, and show the dependencies between them. The definition of the micro frontends architecture is looser, so there are multiple implementations. Typical micro frontends implementation schemes are:

1) Route distribution, which distributes different services to different frontend applications through routing. It is usually implemented through the reverse proxy of the HTTP server, or the routing provided by the application framework. This is the simplest implementation method, and the disadvantages are also obvious. Because the data in different modules is different, it may take longer to load the modules. It is suitable for systems that are difficult to be upgraded, or do not plan to spend a lot of effort to remodel. During the implementation, the code can be returned gradually from the backend to ease the problem of waiting for loading;

2) <Iframe> embedding. Iframe is a very "old" and ordinary technology, which is one of the ways to combine applications of the browser. The system embeds each sub-application into its own iframes, which enables each application to use any framework they need, and applications can run independently without coordinating tools and dependencies with other applications. The biggest advantage of using iframes is that it can isolate the runtime environment of components and applications. Therefore, system development can take a completely different frontend framework, develop a part in React, a part in Angular, and then use native JavaScript or any other technology to develop other parts. As long as each iframe comes from the same source, the messaging between them is fairly simple and powerful. The disadvantage is that you cannot extract public dependencies at build time, bundles will be very large, and it is difficult to support multiple nested iframes;

3) Web Components, which is a set of different technology suites, including custom elements, Shadow DOM, HTML templates and HTML import [16]. By creating reusable custom elements with encapsulation capabilities and using them in web applications. Web components can be imported into web applications in a very elegant way. However, it currently has some difficulties. For example, the system architecture is more complicated, the communication between the components becomes a particularly big trouble.

The main goals of the frontend are simple UI and reasonable business design, making users more willing to use it. Considering the robustness of the system, we think hybrid technology to be the most appropriate solution. In the implementation of this system, we retained the original school roll system and authorization management system through <iframe> embedding, and developed other applications using Web Components technology. This will accommodate new components and without negatively impacting on frontend development time. Another advantage is that you can use multiple component configurations at runtime, providing a dynamic and flexible interface.

The implementation of the micro frontends means the separation of applications. The applications division must be systematically considered application autonomy and single responsibility principle, otherwise it may bring a worse user experience. The model proposed in this paper designs application splitting based on bounded context to divides the context of sub-applications, which will have a better user experience than simply splitting by business.

3. Application of micro frontends in educational management information system

In order to prove the advantages of the micro frontends in the educational management information system, we practiced it on the graduate information system of East China Normal University.

3.1. Splitting the system based on bounded context

Domain-Driven Design (DDD) is mainly used to solve the business analysis of software development when facing complex problems. By introducing a series of concepts such as domain model and common language, software developers and business personnel can clearly define the problems in the same space, and ensure the close relationship between software construction and business model. In this analysis method, the concepts of problem domain, bounded context, and aggregate root are proposed to segment the business domain [17][18]. This paper will use this method to divide the application of the system.

First, model the business domain according to the system's requirements and implementation, and decompose the business of the required services and divide the business boundaries to form a "high cohesion and low coupling"

business subdomain. The system is divided into three business entities: school roll business, student cultivation business and system management business. The following will take the system management business as an example to introduce the service partition design in detail. The system management business determines the boundaries according to the responsibility model to form different business subdomains. The system management business is divided into an authorization domain, a metadata domain, a data center domain, and a notification domain. The metadata domain is the core domain, and each business domain contains the corresponding context. When the business domain is created and the architecture is initially formed, services are identified and modeled as services based on business processes. For example, in the data analysis domain, the system builds a data information center to provide unified information management and analysis. After the modeling is completed, enter the project business goals, business process models, business domain definitions, and bounded contexts, analyze the business from three aspects: processes, data, and rules, and gradually form a service catalog. The detailed design is shown in Fig. 4.

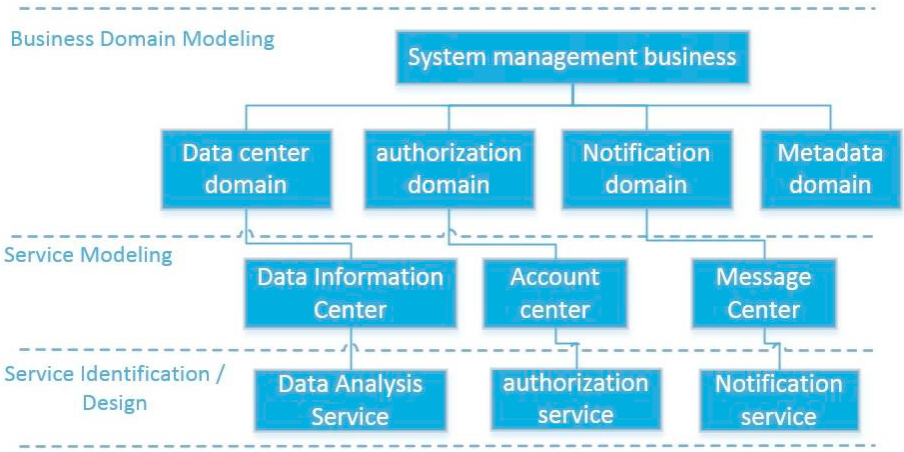

Fig. 4. System Management Domain Design Process

Fig. 5 shows the analysis results of the notification service using the domain-oriented design concept. Each dashed circle in the figure represents a problem subdomain of the problem domain to be solved. The same solid circle represents a bounded context. It should be noted that a domain model often contains multiple meanings, such as the concept of user in the authorization domain. There are differences between different domains, and different areas are not the same. DDD introduces a bounded context, which clearly defines the domain model, domain events, its services, and external dependencies in that context. Fig. 5 shows entities, value objects, aggregates, and aggregate roots in the bounded context of the notification subdomain. In the domain model of the notification subdomain, the message module is used as an entity to distinguish different business modules that need to trigger message notifications. For example, the different types of student transaction requests and the different types of transaction approvals can form a message module. The target users that need notification and the message modules sent by users may be different. A message module can contain multiple target users, and each target user can specify a role ID and an extraction path for extracting Sno / Staff ID and department information from the data provided by the current field event. Therefore, when developers and business personnel are discussing "a postponed transaction notification needs to be sent to *students* and *the department secretary* after the approval", even though "*student*" and "*the department secretary*" are also role names in the authorization subdomain, but they represent the target user in the transaction subdomain. For each notification user, we can define multiple message templates, and each message template can correspond to the notification content when a certain type of transaction request has different approval status. Finally, the placeholder entity corresponds to a placeholder that can be used in the content of a message template defined under a message module. For example, a message template might be "Dear ${name} classmate, your transaction application has been processed, please be patient!", ${Name} is a placeholder for both the template rendering engine and domain experts. The placeholder entity contains the placeholder name, placeholder symbol, placeholder path, and notes. The

placeholder name is the display name, such as "name", the placeholder symbol is the symbol used in the template, such as $name, the placeholder path is used to extract data from the current field event, and notes are used to explain the meaning and use of this symbol. Finally, the user entity is used to indicate a student or teacher in the notification subdomain. It maintains some status information of the current user in the notification service, such as the number of emails that have not been sent, and the time when the notification was last sent. Associated with the user is a message entity, which records information such as the content, creation time, sending mailbox, and sending status of a message.

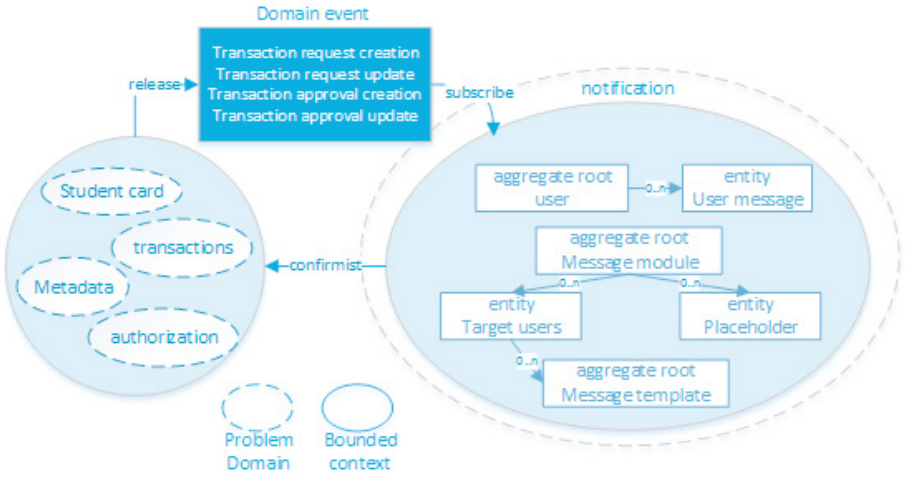

Fig. 5. Bounded Context Analysis Results of Notification Service

Finally, the relationship between the notification context and the school roll context in the context graph given in Fig. 5 is conformist, which means that the notification context not only depends on the school roll context, such as the dependency of the domain model, services, etc., but also expresses notification services on teamwork only accommodate enrollment services, its own code must be adjusted according to changes in the student roll business. At present, the notification context subscribes to domain events in the four fields of the student card, transactions, metadata, and authorizations by system, and use the communication information query service exposed by the student card service. The service accepts parameters as role ID, Sno or Staff ID, and department code. It can return a list of accounts, and each account also contains communication information such as email. By processing these domain events, the notification service obtains the corresponding target users and message templates through the information module entity, generates corresponding notification content based on these templates, and adds corresponding message entities for the user entity.

3.2 Architecture Design of Graduate Information System

The business division of the frontend engineering of graduate information system through the bounded context has three business modules, eleven subdomains, and several sub-services, as shown in Fig. 6. The system performs independent decoupling design on the subdomain, and develops corresponding sub-applications based on the sub-domain. In the development of the new system, the original school roll management service and authorization management service of the system are retained. They are developed through the traditional MVC method, embedded in the <iframe> container, and used *Window.postMessageAPI* to interact with other applications. Use Web Components to develop new sub-applications, mainly using the Angular framework. The system achieves independent deployment and independent development between applications, and has no major negative impact on the user experience. It can also support the use of new technologies to increase system services in the future.

The frontend UI of the graduate information system mainly adopts the style of the Twitter Bootstrap framework, and the backend service is implemented on the Spring Boot framework. Nginx is used in the deployment to implement

load balancing for user access. In the aspect of data, the system uses relational database MySQL for storage. The entire system Use Git to implement source code version control, GitLab acts as a centralized server to host the source code of all business self-applications, and completes the continuous integration and release of the code through Jenkins. The implementation of the project will be explained in detail below.

The system will register accounts for all students, teachers and graduate school administrators to log in [19]. Therefore, the system has designed the master module service of the graduate school. This main application is the unified entrance of the system, the foundation and core of the system, and sends all system requests to the corresponding services. And it will undertake the core business functions of the system, such as account login and authority management, system navigation menus and routing management. The system will also maintain the application registry through this main application to manage other sub-applications.

The system integrates sub-applications under the four major interface modules of school roll management, student cultivation management, data management services, and allowance management according to business attributes, and generates navigation calls to different sub-applications. Later, new sub-applications will be developed based on business needs. The system architecture design is shown in Fig. 7 When the system is running, the main application of the program will obtain our application configuration in the system server, then create sub-applications and bind the life cycle for them. In order to provide users with a better experience, we have designed and built shared UI basic components, and extracted common library dependencies, and directly loaded them through the CDN on the page, which can reduce the loading of the same basic dependencies between different applications, thereby reducing resources requests to optimize system performance.

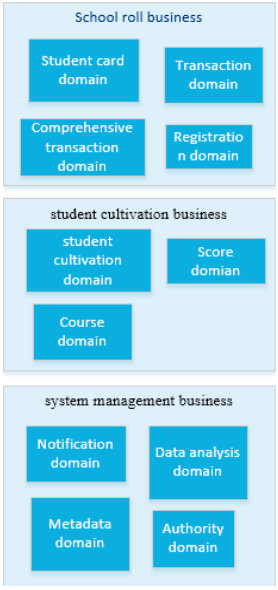

Fig. 6 Domain division of graduate information systems

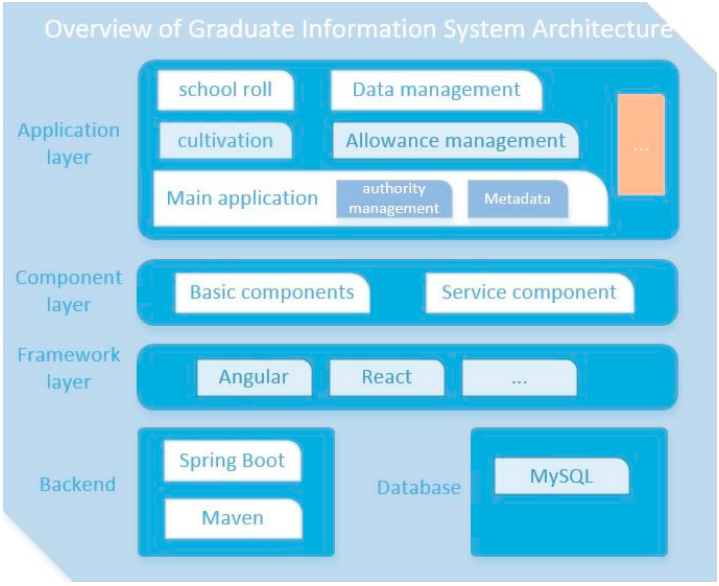

Fig. 7 Overview of Graduate Information System Architecture

3.3 Implementation of graduate system based on micro-frontend

3.3.1 System application registration

The system creates a Json configuration file for sub-application registration. Each sub-item corresponds to the information of a sub-application, including the application name, routing prefix, and project entry file. This configuration file needs to be updated when sub-applications are added or deleted. The main application does not need to be repackaged, and directly reads and changes the configuration file. When the system is started, the main application of the system loads the configuration file, then registers sub-applications such as school roll services, cultivation services, and data management services through the registration application interface. Finally, the main

application executes *start*() to start the sub-applications, and generate the system navigation menu bar. When the sub-application is started, the graduate information system activates the corresponding sub-application through the matching function of routing information and loads the code.

3.3.2 Application communication

Although the communication between applications is reduced as much as possible through the division of services, some level of cross-app communication is often needed. The system uses the Event mechanism to design the *EventHandler* class to communicate with the publish and subscribe, Under the Event mechanism, the application that publishes the event does not need to be concerned with who is subscribing to this event, and the application that subscribes to the event does not need to know the source of the event. *EventHandler* member variables *events* are used to store event handler functions, and store event names and functions in key-value pairs. The communication process is as follows: First, an *EventHandler* instance is created when the application is registered; then, the application completes communication by publishing and subscribing to custom events after startup; finally, the child application unsubscribes from events before uninstalling.

Under the development of front-end and back-end separation, the web front-end and server may cause cross-domain issues. The system uses the current mainstream cross-domain resource sharing (CROS) mechanism.

3.3.3 Graduate Information System

At the business layer, the graduate information system has designed different functional components for students, teachers, and graduate school administrators, in which students only need to access personal student status information, teachers need to manage their guidance students and class information, and graduate school administrators manage all student training processes, system data and account permissions. The system's final implementation page is shown in Fig. 8.

When the system is launched, the configuration file is loaded and the <Link> elements that navigate to the specific micro frontends application are rendered. When the user logs in to the system, the route is changed according to the user's click on the navigation bar <Link>, and the system activates the corresponding sub-application according to the route change. Fig. 9 shows the application interaction of graduate information system. The solid line indicates that the user clicks the application link. When the user clicks the link, the current route will be changed. When the route is changed, the system will render the corresponding application. For example, when the teacher clicks cultivation management on the homepage, the route will be switched from "/yjsy/" to "/yjsy/py/", the system will render to the cultivation management application, and fetch the data of the cultivation management application from the server. When the system needs to add a new sub-application, it is only necessary to complete the development and deployment of the sub-application components, and then modify the configuration file to use it in the system.

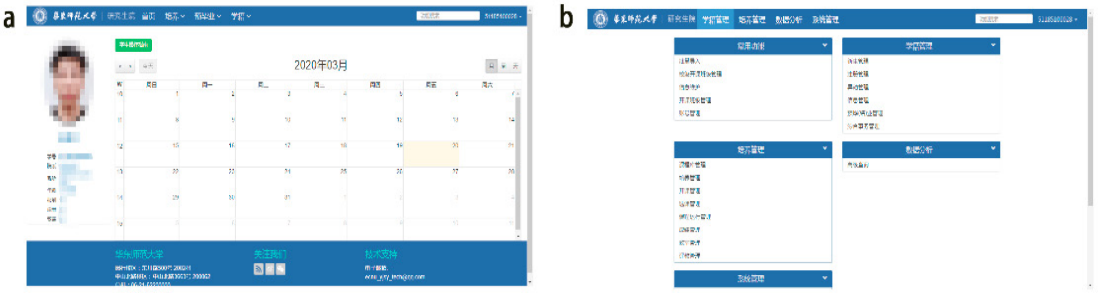

Fig. 8 (a) Students' page; (b) Administrator page;

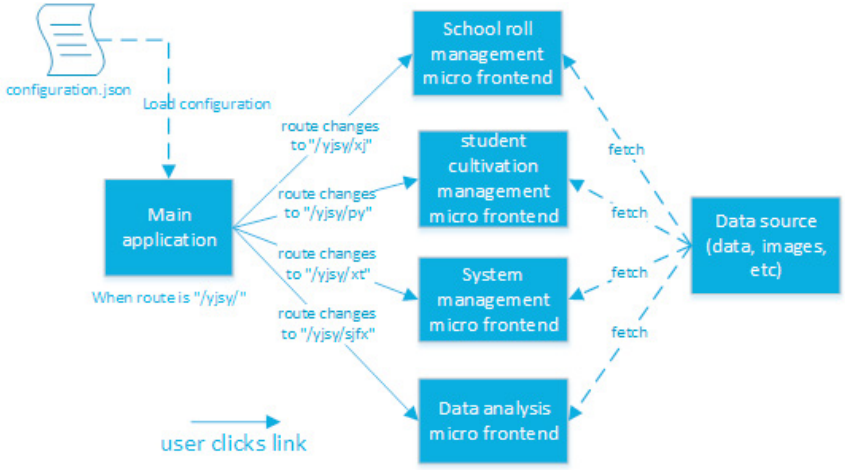

Fig. 9 The application interaction of graduate information system

4. Related work

Since Martin Fowler has clearly defined microservice, microservice have attracted widespread attention. Its advanced ideas have been deeply rooted among many architects and engineers. Yi Wang has proposed methods for REST-based service-oriented construction of educational management information system [20], but it did not make a new attempt on the frontend. Holger Harms analyses the strategies for organizing and implementing frontends, and he considers self-contained systems (SCS) to be a better approach for frontend [12]. However, most systems are still being developed with traditional architectures, and there is no design of SCS. It is difficult to meet the new requirements of the new era in terms of technology and resources.

The idea of micro frontends based on the microservice has been proposed, and has attracted the attention of some scholars [15]. It has proposed single-spa, Mooa and other model designs. But single-spa is complex to build and does not support the separation of deployment of different applications. And Mooa did not consider to isolate JS, avoid CSS conflicts, and load resources on demand. Based on the micro frontends, this paper designs the DDD-based self-contained system, and designs a complete implementation scheme for the graduate information system.

5. Conclusion

In order to better adapt to the agile development of education information system, this paper proposes a micro frontends architecture and applies it to the graduate information system of East China Normal University. Based on the micro frontends architecture, the system designed a new business model based on the bounded context, fully meeting the new business requirements, technically solving the problems of slow response and insufficient functions of the original system, and providing a better experience for teachers and students. The design of the micro frontends reduces the degree of coupling between the systems, enables functional development to be completed independently, and makes good planning and design for future development work. It increases the composability and scalability of the educational management information system, and greatly reduces the further cost.

In the future, we will gradually design more norms and standards of the framework, optimize development process, and reduce the complex integration of the system. We will also make further optimizations in system performance, and continue to explore its practical applications.

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] Education Informatization 2.0 Action Plan, (2018) BeiJing: Ministry of Education of the People's Republic of China.

[2] Jia, You, and G. Lee. (2019) "Current Situation and Discussion of Teachers' Information Literacy in the Era of "Education Informatization 2.0"." *Itm Web of Conferences 26*.

[3] L. Wang, P. Wan, G. Xu and J. Ma, (2010) "Design and implementation of management system based on educational technology" *2010 3rd International Conference on Computer Science and Information Technology*. IEEE, 9: 503-507.

[4] S. Lingjuan and Z. Yanqing. (2014) "Application of Computer Technology in Sports Teaching Management of Colleges and Universities." *2014 Fourth International Conference on Instrumentation and Measurement, Computer, Communication and Control*. IEEE: 161-164.

[5] Liu, Zhongde, and Yongqing Yang. (2016) "Research on the Construction of Educational Administration Management Information in Colleges and Universities." *International Conference on Education, Management, Computer and Society*. Atlantis Press.

[6] TANG Lu-min, YU Ruo-nan, DONG Qi-wen, HONG Dao-cheng, FU Yun-bin. (2018) "A review of non-intrusive sensing based personalized resource recommendations for help-seekers in education." *Journal of East China Normal University (Natural Science)*, 5:17-29.

[7] Yanling Jin. (2014) "Educational Administration Management System and Modern Education Management on the Perspective of Modern Information Technology." *Information Technology Journal*, 13: 340-346.

[8] N. Dragoni, S. Giallorenzo, A. L. Lafuente, M. Mazzara, F. Montesi, R. Mustafin, L. Safina. (2017) "Microservices: yesterday today and tomorrow" *Present and Ulterior Software Engineering*, Springer, pp. 195-216.

[9] Hamidi, Farideh, Meshkat, Maryam, Rezaee, Maryam, and Jafari, Mehdi. (2011) "Information Technology in Education." *Procedia Computer Science 3*, 369–375.

[10] Yunpeng Cai, (2010) "The research and application of SaaS in educational Information system based on educational metropolitan area network" *2010 International Conference on Educational and Information Technology, Chongqing*, pp. V2-508-V2-510.

[11] Cherniavsky, J.C., Soloway, E. (2002) "Editorial—A Survey of Research Questions for Intelligent Information Systems in Education." *Journal of Intelligent Information Systems* **18**, 5–14.

[12] Harms, Holger, Collin Rogowski, and Luigi Lo Iacono. (2017) "Guidelines for adopting frontend architectures and patterns in microservices-based systems." *Proceedings of the 2017 11th Joint Meeting on Foundations of Software Engineering*.

[13] S. Ivanova and G. Georgiev, (2019) "Using modern web frameworks when developing an education application: a practical approach " 2019 42nd International Convention on Information and Communication Technology, Electronics and Microelectronics (MIPRO), Opatija, Croatia, pp. 1485-1491.

[14] Vallecillos, Jesús, et al. (2014) "Dynamic mashup interfaces for information systems using widgets-as-a-service." *OTM Confederated International Conferences" On the Move to Meaningful Internet Systems"*. Springer, Berlin, Heidelberg.

[15] Yang, Caifang, Chuanchang Liu, and Zhiyuan Su. (2019) "Research and Application of Micro Frontends." *IOP Conference Series: Materials Science and Engineering*. Vol. 490. No. 6. IOP Publishing.

[16] Wusteman, J. (2019), "The potential of web components for libraries", *Library Hi Tech*, Vol. 37 No. 4, pp. 713-720.

[17] Rademacher F., Sachweh S., Zündorf A. (2018) "Towards a UML Profile for Domain-Driven Design of Microservice Architectures." *International Conference on Software Engineering and Formal Methods*. Springer, Cham

[18] Rademacher F, Sorgalla J, Sachweh S. (2018) "Challenges of domain-driven microservice design: a model-driven perspective." *IEEE Software*, **35(3)**, 36-43.

[19] Shimozono, Koichi, et al. (1998) "User management in an educational computer system: personal information management." *Proceedings Twelfth International Conference on Information Networking (ICOIN-12)*. IEEE.

[20] Yi Wang and Li Zhao, (2010) "Service-oriented Educational Management Information System construction" *2010 International Conference on Computer Application and System Modeling (ICCASM 2010)*, Taiyuan, pp. V7-554-V7-557.