DESIGN AND IMPLEMETATION OF A GENERAL ENGLISH EXAMINATION SYSTEM BASED ON WEB

[1]FENG GAO, [2]JIAHUI QI, [3]YAN YE, [4]MIN WU

[1,2,3,4]Center of Modern Educational Technology, University of Science and Technology of China, China
E-mail:[1]fenggao@mail.ustc.edu.com, [2]jhqi@ustc.edu.cn, [3]zyyp@ustc.edu.cn, [4]minwu@ustc.edu.cn

Abstract— To solve the limitations of single exam mode and fixed question-types of most existing English examination system, the paper put forwards a kind of generalEnglishexamination system based on Web with diversified exam modes and extensible question-types. The system is compatible with four kinds of exam modes, the way of invigilating and students' examination process of different exam mode is different. The feature of extensible question-types is implemented through component-based structure design of question-types, using the JSON data format and React technology. It has been proved by using that the system can applied to various types of common exam in English teaching and testing, can meet the diversified needs for the question types of different users and can improve utilization and generality of the system greatly.

Keywords— General Examination System, Exam Mode, Question-Type Structure, Extensible Question-Types, Componentization.

I. INTRODUCTION

With the popularization of the network and the rapid development of computer-related technology, the use of computer-related technology in English teaching and testing has been widespread concernedby researchers from all walks of life. Computer examination system has been widely used because of its special advantages, such as improving scoring efficiency, reducing teachers' labor intensity, shorteningthe cycle ofresult statisticsand other advantages[1].

Through research and analysis of the existing English examinationsystem,it is found that most English examination system has some limitations. It manifests in the following two aspects: in aspect of exam mode, the majority of examination systems only support one single exam mode, such as independent practice mode or strict exam mode; in terms of question-types, the majority of examination systems provide only a few fixed specific question-types involved in the examination[2-4], it is difficult to meet the different system users' flexible demand of question-types. Because of these two limitation, most English examination system cannot apply to different kinds of examination or cannot meet the needs of system users. It reduces the usage and generosity of the system.

In response to these two limitations talked above, the paper realize the general English examination system. It is compatible with four kinds of exam mode and it implements the extension of the question-types by using the React technology. So the generosity of the system is improved. The authors hope to provide a viable universal idea for the development of other similar examination system through the implementation of this system.

II. RELATED WORK

2.1. Selection of Data Exchange Format

JSON is easier to read and code compared to XML and can be parsed by the browser supported JavaScript and can easily complete transformation between JSON objects and JSON strings[5-6]. These peculiarities make JSON be the ideal data exchange format in Web Applications.Therefore, the author selects JSON format as the data transmission carrier.

The data in JSON format needs to be transformed in different states in the transmission process. The serialization of JSON data is the process of turning JSON objects into JSON strings which can be saved or transmitted. Conversely, deserialization is the process of transforming the strings flow into the object.

In the browser's JavaScript syntax environment, we can turn JSON object into string through the method "JSON.stringify" and complete the process of deserialization through "JSON.parse".

2.2. React Technology

The implementation of the extension of the question-types uses the modular thinking of the React technology. Each component contains separate logic and UI, easy to understand and maintain. And the components can be combined or nested with other components to improve the reusability of components. The extension of the question-types is implemented by visible nested and combination of components based on the modular question-type structure.

In this process, frequently, changing data needs to be reflectedon the UI via the DOM real-timely operation. But complex and frequent DOM operation will reduce the performance of Web applications to some extent[7].

The Virtual DOM mechanism has been introduced in React[8]. In React, all DOM operations are carried out via Virtual DOM. When the entire data changes, React will obtain the difference of the DOM through comparing with last DOM. Then it updates the actual

change to reconstructs the entire DOM tree. Therefore, the improvement of performance is achieved not only, but also the render of the entire UI is the only thing to consider by the developers, rather than focusing on how to update one or more specific DOM elements as some data changes. The working principle of Virtual DOM is shown in **Fig.1**.

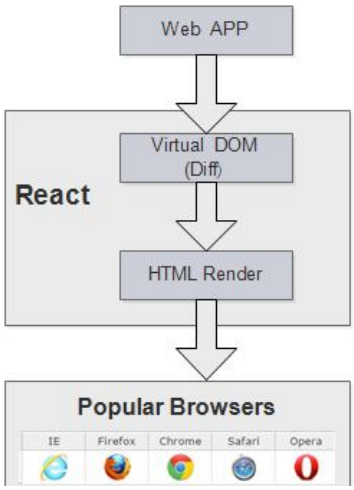

Fig.1. Working Principle of Virtual DOM

2.3. Requirements Elicitation

The extraction, negotiation and documentation of the stakeholders' requirements is completed in use of the requirements engineering technology. The requirements engineering framework which is used in the system is shown in **Fig.2**.

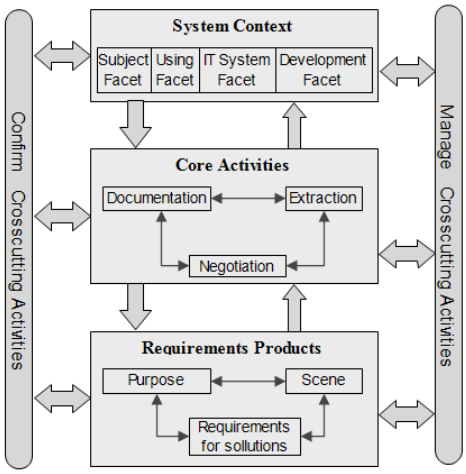

Fig.2. Requirements Engineering Framework

The functional requirements of the system obtaining through the requirements engineering techniques include: systems management, extension of question-types, questions management, paper management, examination transaction management, supports of kinds of exam mode, real-timing examination, teachers invigilation, exam scheduling and monitoring, paper scoring, results querying and statistics, and self-trainin. The performance requirements include: fast rendering of the paper structure, real-time interaction between teachers and students, the speed of the paper loading and the answers uploading, support of cross-platform and generosity and so on.

III. DESIGN OF SYSTEM

The functional modules of the system are designed and realized via the B/S architectures. The database server is the MySQL relational database management system. The development of backend bases on the PHP framework ----Symfony2. And the implementation of the front end combines three kinds of web development technique: JavaScript, CSS and HTML.

The overall architecture of the system is shown in **Fig.3**:

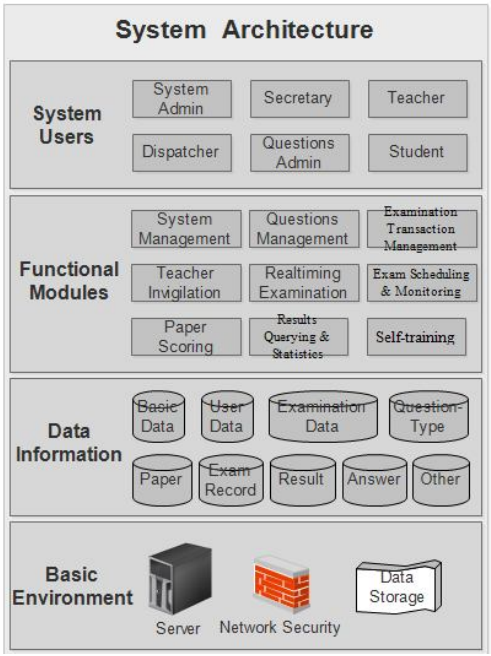

Fig.3. Overall Architecture of the System

Compared to the traditional examination system, the main innovations of this system is the detail division of the users'roles, the compatibility of four kinds of exam mode and the extension of the question-types. Although the correlated functions of the system will varies in some degree because of the innovations, they are also similar to the functions of traditional examination system. The implementation of the three innovations will elaborate in detail in the follow part of the paper.

IV. IMPLEMENTATION OF THE SYSTEM

4.1. Division of the Users' Roles

Hierarchical division of system users' roles can make the functional responsibilities of every user clear so that the system can be easily managed and used. Then the security of the system access can be ensured

through strict access controls and different user interfaces can be provided depending on the different user role.

4.1.1.Six kinds of roles

Combining with the stakeholders and functions of the system, the roles of the users are divided into the following six kinds:

ROLE_ADMIN: ensure the normal operation of the system, maintain the basic data(grade information, student information, exam information) and backup and restore the server data; In addition, the users who has the role of ROLE_ADMIN have all the other users' rights (except students).

ROLE_SECRETARY: manage and maintain the teachers' and students' information, manage exams, arrange the examination transaction and scoring and archive the examinations.

ROLE_TEACHER: be responsible for the invigilation and scoring of the examination.

ROLE_DISPATCHER: be responsible for dispatching in the process of the examination (update or add students information, etc.) and monitoring the examination room status.

ROLE_QUESTION_ADMIN: be responsible for management of question-types, questions and paper.

ROLE_STUDENT: participate in examination or self-training system and query result.

4.1.2.Roles levels

There is some overlap in the roles of the users, such as the ROLE_ADMIN has the right of the ROLE_SECRETARY, the ROLE_DISPATCHER and the ROLE_QUESTION_ADMIN. Instead of associating many roles to users, we can define role inheritance rules by creating a role hierarchy provided by the security mechanisms of Symfony2 framework so that some roles can automatically have other roles[9]. Specific codes are as follows, you can specify the ROLE_ADMIN have three other roles:

```
role_hierarchy:
    ROLE_ADMIN: [
        ROLE_SECRETARY,
        ROLE_DISPATCHER,
        ROLE_QUESTION_ADMIN
    ]
```

4.1.3.Rights access control

In the process of using the system, the system can control the access to the functional modules through the division of the users' roles. We can specify the protect URL patterns and the roles allowed to access the URL through configuring "access-control" in the "security.yml" file. Code examples are as follows:

```
access_control:
    - {
        path: ^/manage/academy,
        roles: ROLE_ADMIN
    }
```

4.1.4.Roles in templates

In show of the front end template (the twig template is used in this system), different content and function can be presented according to the user's role. In this case, the method "is_granted" can judge the roles of the user in templates. For example:

```
{% ifis_granted('ROLE_ADMIN') %}
    <a href="…">Delete</a>
{% endif %}
```

4.2. Support of Four Kinds of Exam Mode

Combined with the limitation of single exam mode of the existing examination system, the system is compatible with four kinds of exam modes. These four kinds are as follows:

1. strict mode: The examination place of strict mode was fixed. The examination process(include the beginning and ending of the examination) is controlled by the invigilator.

2. non-strict mode: The examination place of non-strict mode was fixed. The invigilator just needs to control the entrance of the student within the specified time. It is not necessary to begin or end the examination at the same time. The students can start at a specific period of time after admission.

3. homework mode: The examination place is not fixed. There is no invigilator. The students can complete the whole examination via the appropriate URL after the examination arrangement by teachers

4. self-training mode: The student take an examination according to their own need in the self-training system.

The examination secretary can specify one of the exam mode via the examination management function in the examination transaction management module. Different flow control is provided in strict mode examination and non-strict mode examination. The method "getExamMode" can get the mode of the current examination. Then the different UI can be provided to the teachers and the students according to the exam mode. In terms of the other two kinds of exam modes, the students can complete the examination independently without the invigilator. The specific examination process of four kinds of exam mode is shown in **Fig.4**.

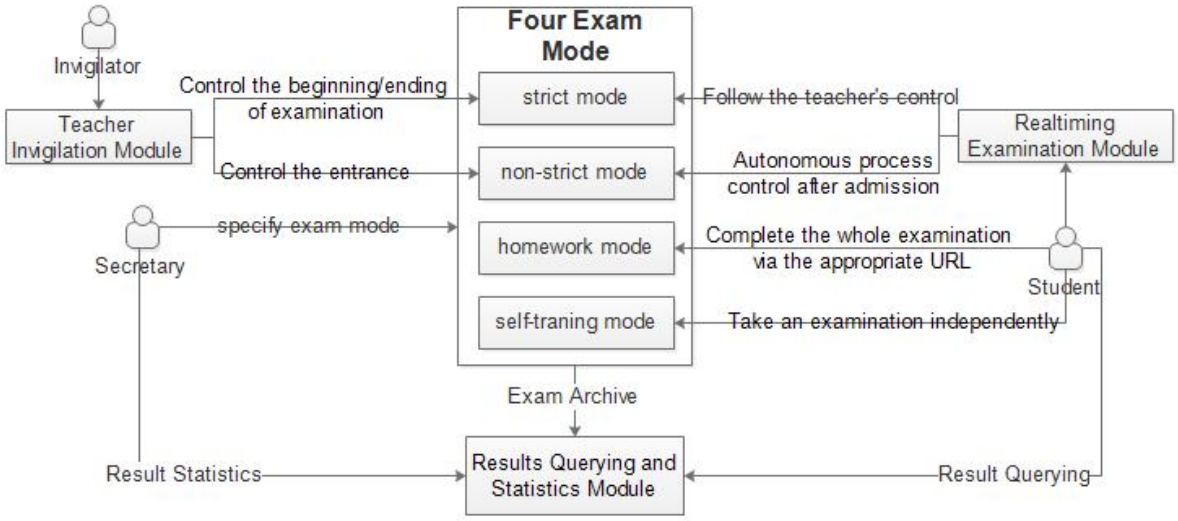

Fig.4. Examination Process of Four Kinds of Exam Mode

4.3. Implementation of the Extension of Question-types

4.3.1. Structure of question-type

There are 9 properties included in the structure and interaction of the common question types through splitting, detailing and abstracting. The JSON is used to store and exchange these properties. These 9 properties are defined in **Table 1** as follows.

The "questions" propertyof the question-type is an array. Each property of the array is a question. Each question also has 6 properties shown in **Table 2**.

Table 1: The Properties in the Structure of the Question-type

Name	Type	Description	Default
stem	String	item stem	""
show-stem-length	Boolean	whether or not show the length of item stem	false
flowable	Boolean	whether the item is flowable	false
shuffle	Boolean	whether or not shuffle the options	false
show-options-order-num	Boolean	whether or not show the order number of options	true
options	Array	item options	[]
questions-num-limit	Boolean	whether or not limit the number of the questions	true
pre-show	Boolean	whether or not pre-show the questions	true
questions	Array	questions including in the question-type item	[]

Table 2: The Properties in Question

Name	Type	Description	Default
stem	String	question stem	""
pause	Integer	the duration of the answer time for the flowable question	0
options	Array	the options of the question	[]
strict	Boolean	whether or not the answer exactly match the reference-answer	false
reference-answer	String/Interger	The reference answer of the question	""
answer-analysis	String	The analysis of the answer	""

4.3.2.Component structure of question-type

The author rethink the UI of the extension of question-types combining with the componentization provided by the React technology. Each part of the UI which has an independent function is defined as a components. Then the bigger components can be consist of the smaller component through combination and nested. Finally, the whole UI of question-type extension can be completed. The component structure of the question-typeis shown as follows in **Fig.5**:

Question-type component includes three parts: question-stem component, options component, questions components. These threesub-components has their own child components, such as parameters-setting component, question component and so on. It is easy to complete the construction of the whole question-type component through the combination and nested of each child component via visible UI operation including adding question-stem and adding questions.

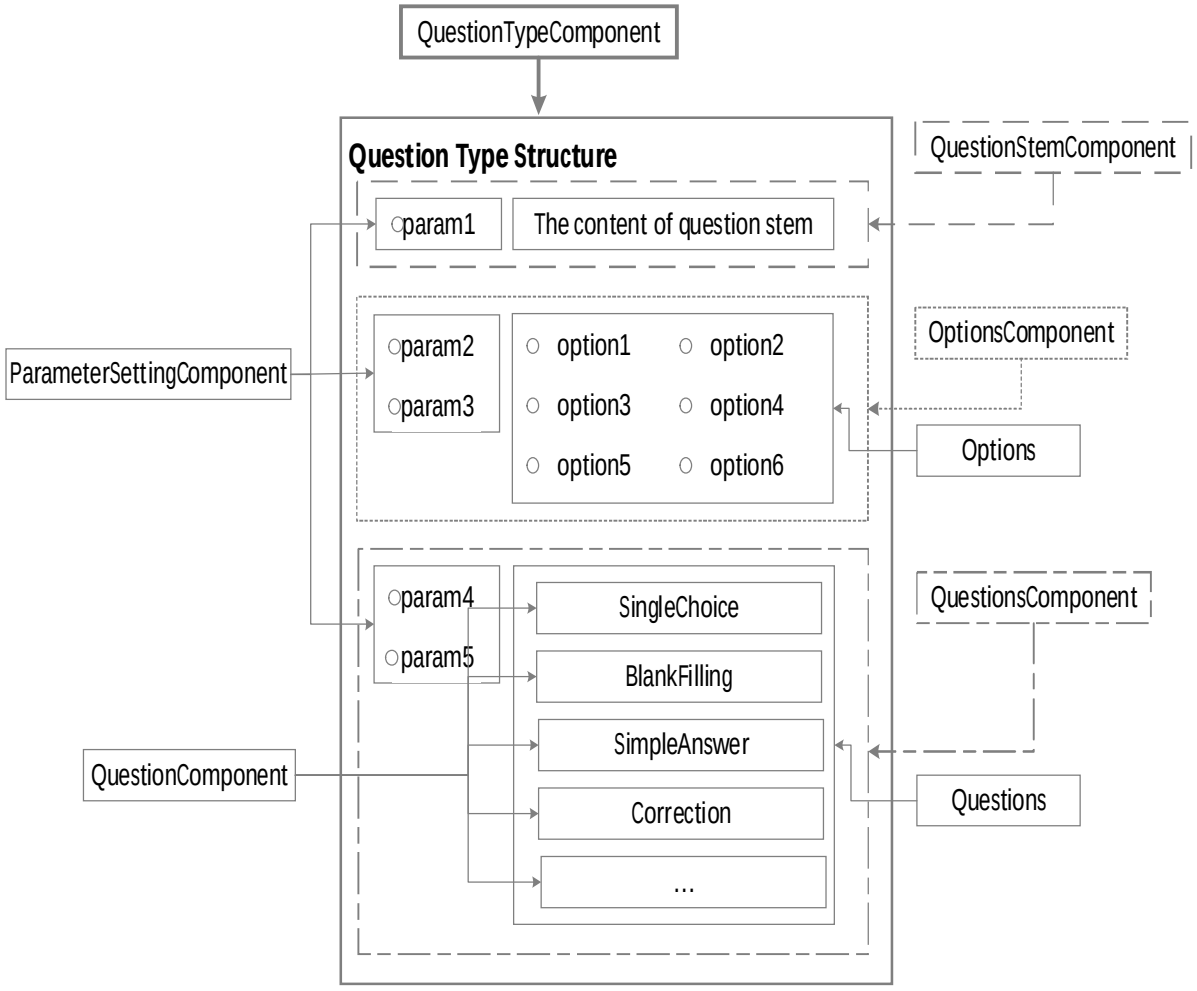

Fig.5. Component Structure of Question-type

4.3.3.Implementation of extension of question-type

On the basis of component structure of the question-type, the method "React.createClass" provided by React can use to create the component class. Code is as follows:

```
var QuestionType = React.createClass({
    render: function(){
        //output the component
        //the return value is the DOM element
create by "React.createElement" method
    }
});
```

In this system, all the state of the component involved are stored in question-type component, the state can be initialized by the "getInitialState" method. And the author has defined the method"handleChangeState" to modify the state of the question-type componentand store the serialized state. This method can be pass to the child components in the process of nesting and combining. When the state of child components changes, it runs to set the state of the whole question-type component. Code is as follows:

```
handleChangeState: function() {
    this.setState({
        item: item
    });
    $item_structure.val(JSON.stringify(item));
}
```

The **Fig.6** is the UI of question-type addition.In this web page, the users can enter basic information of question-type and add or delete child components to complete the extension of question-types.

CONCLUSIONS

In this paper, a general English examination system based on Web which is compatible with four kinds of exam mode and achieves the extension of the question-types is designed and implemented.

Although this paper provides a viable universal idea for the development other similar examination system through the implementation of this system, but the realization of a general examination system also need to consider some other content. The authors suggest that an examination system which is multidisciplinary can be the next study in order to enhance the versatility of the examination system further.

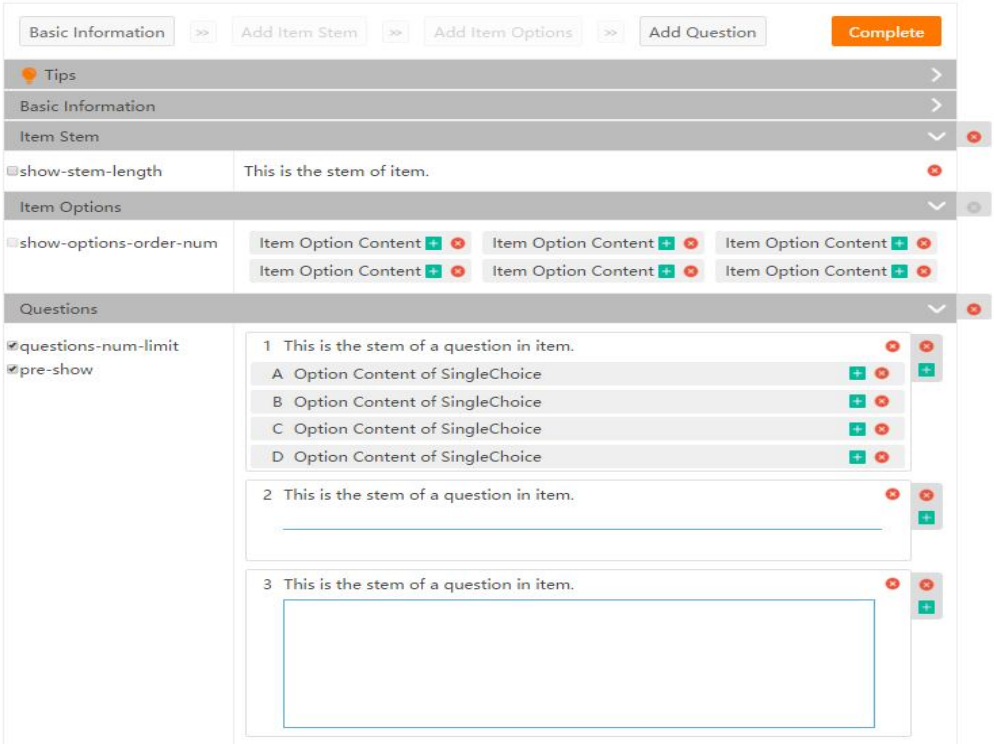

Fig.6. UI of Question-type Addition

REFERENCES

[1] CAI Bi-ye. Implementation of an Examination System Based on Web[J]. Computer Engineering and Science, 2002, 24(4): 31-34.

[2] Jian Miao. Items and Papers Design and Implementation in Computer-based Oral Testing System [D]. Hefei : University of Science and Technology of China, 2008.

[3] WANG Hao, ZHANG Wen-jing. The Evolution of CET-4 and CET-6[J].Overseas English, 2014, (12): 65-67.

[4] Yunqina Wang. Study on test Item Types in computer-based oral English testing system[D]. Hefei : University of Science and Technology of China, 2008.

[5] ZHANG Tao, HUANG Qiang, MAO Lei-ya, GAO Xing. A algorithm of object serialization based on JSON[J]. Computer Engineering and Applications, 2007, 43(15): 98-100.

[6] Chen Wei, Jia Zongpu. Research on Reducing XML's Data Redundancy with JSON[J]. Computer Applications and Software, 2012, 29(9): 188-206.

[7] Introductory tutorial of ReactJs[EB/OL]. 2015-05-08. http://www.cnblogs.com/yunfeifei/p/4486125.html.

[8] A JavaScript library for building user interfaces | React[EB/OL]. http://facebook.github.io/react/index.html.

[9] The SymfonyBook: security [EB/OL]. http://symfony.com/doc/current/book/security.html

★★★