

# Design and Implementation of the Bus Information Management System

Songhao Li<sup>1, a \*</sup>, Chao Wang<sup>1, b</sup>

<sup>1</sup>School of Computer and Software Engineering, Nanyang Institute of Technology, Nanyang, China

<sup>a</sup>lisonghao@nyist.edu.cn, <sup>b</sup>wangchao@nyist.edu.cn

**Abstract.** With the accelerated pace of urbanization, urban transportation has become increasingly complex, and residents have an increasingly urgent demand for convenient and efficient bus query services. This study aims to design and implement a bus information management system based on the Spring Boot framework to improve the convenience of obtaining bus travel information. The system adopts the B/S architecture, with the front-end building the user interface using HTML, CSS, JavaScript and the Vue.js framework to achieve friendly interaction with users; the back-end is based on the Spring Boot framework, combined with the MyBatis persistence layer framework to operate the MySQL database for data storage and management. The system has core functions such as line query, station query, and transfer query. Line query supports users to quickly obtain detailed information of the line by entering the line name, including passing stations, first and last bus times, fares, etc.; station query allows users to enter the station name to query all buses passing through the station; transfer query intelligently plans the optimal transfer scheme according to the starting and ending points entered by users. After testing, the system has achieved the expected goals in terms of function realization, performance and user experience, providing strong support for urban residents' bus travel and having high practical value and promotion potential.

**Keywords:** Spring Boot Framework; Vue.js; MyBatis; Bus Information Management; B/S Architecture

## 1. Introduction

With the rapid growth of urban population and the continuous expansion of urban scale, problems such as traffic congestion and low travel efficiency have become increasingly prominent, which seriously affect residents' daily travel and the sustainable development of cities. Public transportation, as a core part of urban traffic, its informatization construction has become a key path to solve the above problems. Traditional bus information query methods such as paper bus maps and station query boards have defects such as untimely information update and poor query convenience, which can no longer meet the travel needs of modern residents [1].

The Spring Boot framework, as a lightweight Java development framework, greatly simplifies the project construction and development process with its automatic configuration and starter dependency characteristics, and improves the development efficiency and stability of the system [2]. Based on this, this study designs and implements a bus information management system. By integrating the front-end Vue.js and back-end Spring Boot technologies, it builds an information platform integrating bus information query and management. It not only provides efficient bus travel information services for residents, but also provides data support for urban traffic management, which is of great practical significance for alleviating urban traffic congestion and promoting the sustainable development of cities [3].

In the international context, developed countries started early in the field of bus informatization, and their technical systems have become mature. Metropolises such as New York, London and Tokyo have built a full-domain bus information service platform by integrating cutting-edge information technologies such as GPS and GIS, realizing real-time tracking of bus dynamic trajectories and real-time collection of bus operation data [4]. In China, with the rapid development of Internet technology and the popularization of smart phones, bus information systems have also been widely applied. Major cities have launched their own bus query APPs, and mainstream travel mobile applications have formed technical benchmarking with overseas bus information platforms in the basic functional architecture [5]. However, compared with foreign countries, China's bus information

query system still has gaps in data accuracy, real-time performance and functional richness, which makes the research and development of a more optimized bus information management system necessary.

The main content of this research is to design and implement a bus information management system with B/S architecture and front-end and back-end separation based on Spring Boot and Vue.js frameworks. The system includes two modules: the management end and the user end. The management end is responsible for the addition, deletion, modification and query of bus-related data to ensure the timeliness and accuracy of information; the user end provides users with multiple query functions such as bus lines, stations and transfers. At the same time, a reasonable MySQL database structure is designed to provide solid data support for the system, and the system is fully tested to ensure its stable and reliable operation.

## **2. Related Technologies and Architecture**

### **2.1 Core Development Technologies**

Vue.js is a lightweight JavaScript front-end framework that adopts a component-based development model, which can split the page into independent and reusable components, improving the maintainability and development efficiency of code [6]. Its two-way data binding mechanism can realize real-time synchronization between data and views, allowing developers to focus on writing business logic. In this system, Vue.js is used to build the front-end interface of the management end and the user end, combined with the Element-UI component library to achieve a responsive layout, and the Vue-admin-template template is used to speed up the development of the management background, greatly reducing the front-end development workload.

Spring Boot is an innovative framework based on the Spring framework, whose core goal is to simplify the initial construction and subsequent development process of Spring applications [7]. In terms of configuration, its automatic configuration function can intelligently configure the basic components required by Spring applications according to the project's class path composition and dependency relationship, eliminating the complicated XML configuration in traditional Spring development; in terms of deployment, it has built-in servers such as Tomcat and Jetty, which supports packaging applications into independent executable files for convenient deployment; in terms of architecture, it provides good support for the microservice architecture, reducing the coupling degree of the system. In this system, Spring Boot acts as the core back-end framework, responsible for processing business logic, parsing front-end requests, and interacting with the database efficiently.

MyBatis is an excellent persistence layer framework that realizes the separation of SQL statements and Java code, supports custom SQL and stored procedures, and can flexibly operate the database [8]. MySQL is an open-source relational database with high performance, high reliability and good scalability. Its InnoDB storage engine provides transaction security and row-level locks to ensure data integrity and concurrent processing capabilities [9]. In this system, MyBatis is used to operate the MySQL database to complete the storage and management of core data such as user information, bus line information, and station information.

In addition, the system integrates Spring Security for system security authentication, which encrypts and stores user passwords and prevents network attacks such as cross-site request forgery to ensure data security[10]; mavon-editor, a rich text editor, is used to realize flexible editing of bus information and ensure the richness and readability of information display; the axios library is used for asynchronous communication between the front and back ends to achieve efficient data request and response processing.

### **2.2 Overall System Architecture**

The system adopts a B/S architecture with separated front and back ends, which decouples the development and deployment of the front and back ends. The front end is responsible for building the user interface and processing interaction logic, and the back end focuses on business logic operation, data storage and interface provision [11]. The two communicate through the HTTP protocol, and the back end returns data to the front end in JSON format, which effectively improves the development efficiency and system maintainability.

The front-end layer is based on the Vue.js framework, builds the interfaces of the management end and the user end with HTML, CSS and JavaScript, and realizes asynchronous data requests with the back end through the axios library to ensure the response speed and interactivity of the interface. The back-end service layer takes Spring Boot as the core, divides the business logic layer and the data access layer, processes the front-end requests such as query, add, modify and delete, realizes the interaction with the database through MyBatis, and integrates Spring Security to complete identity authentication and permission control. The data persistence layer uses the MySQL database, designs a reasonable data table structure, stores core data such as users, bus lines, bus stations, bus dynamics and announcement information, and provides data support for the realization of system functions.

### **3. Demand Analysis**

#### **3.1 Feasibility Analysis**

The technologies selected for the system, such as Vue.js, Spring Boot, MyBatis and MySQL, are all current mainstream open-source technologies with mature technical systems and perfect community support, and there are a large number of development documents and cases for reference [12]. The componentization and two-way data binding characteristics of Vue.js can quickly build a user-friendly front-end interface; the automatic configuration and efficient integration capabilities of Spring Boot simplify back-end development; the high performance and scalability of MySQL ensure the stability of data storage. The development team has the development capabilities of the relevant technologies and can realize all functional requirements of the system, so the system is technically feasible.

The frameworks, databases and development tools used in system development are all open-source and free resources, without the need to pay copyright fees, which greatly reduces the development cost [13]. After the system is launched, it can meet the bus query needs of urban residents, improve the utilization rate of public transportation, and indirectly reduce the cost of urban traffic governance. At the same time, the system has a simple structure and low later maintenance cost, and has good economic benefits and application prospects, so it is economically feasible.

The system adopts a simple and intuitive interface design with a clear operation process, and users can quickly query bus information by entering keywords without professional computer operation knowledge [14]. After performance optimization, the system can still run stably under high concurrency scenarios, with fast page loading speed and accurate query results, which can meet the use needs of users of different age groups. In addition, the system provides friendly error prompts and operation guidance, further reducing the user's operation cost, so it is feasible in use.

#### **3.2 Functional Requirement Analysis**

The system is divided into two modules: the management end and the user end, which face system administrators and ordinary urban residents respectively to meet the functional requirements of different roles.

The management end is operated by system administrators, and its core function is the management and maintenance of bus information to ensure the accuracy and timeliness of system data. The specific functions include user management, line and train number management, bus station and information management, dynamic and surrounding service management, system management and comment management [15]. User management realizes the addition, deletion, modification and query of user information, and supports precise search by user account and name; line and train number management completes the addition, deletion, modification and query of bus line and train number information, and verifies the uniqueness of lines and train numbers to avoid data duplication; bus station and information management maintains the basic information of bus stations and the core operation information of buses; dynamic and surrounding service management updates the real-time dynamic information of bus stations and the surrounding service information of stations; system management maintains the basic information of the system and manages the carousel map and announcement information; comment management views and replies to user comments to improve system interactivity.

The user end is for ordinary residents, and its core function is the convenient query of bus information. The specific functions include bus line query, bus station query, bus information query, surrounding service query and announcement information browsing [16]. Bus line query allows users to enter the line name, starting station, terminal station or transfer information to query the detailed information of the line such as passing stations, first and last bus times and fares, and supports line evaluation and collection; bus station query enables users to enter the station name, road name or station location to query the information such as buses passing through the station and arrival time; bus information query allows users to enter the bus number, shift, starting station or terminal station to query the detailed operation information of the bus; surrounding service query provides users with the surrounding service information of bus stations to provide supporting reference for travel; announcement information browsing allows users to view the bus announcements and notifications released by the system, and supports the search, like and collection of announcements.

### **3.3 Non-Functional Requirement Analysis**

System performance indicators are the core factors determining the quality of user experience. In terms of response speed, the system should ensure the efficient and smooth page loading process, and control the delay from the user's request to the complete presentation of the page within a reasonable threshold [17]. For high concurrency scenarios, the system needs to have a strong concurrent bearing capacity, ensure stable response performance during business peak hours through optimizing resource allocation strategies and task scheduling mechanisms, and avoid system performance degradation due to the surge of concurrent requests.

The system adopts the core design principle of modularization, with low coupling between each functional module, which is convenient for adding new functions and expanding data dimensions in the later stage [18]. For example, functions such as real-time bus positioning, code scanning for riding and bus travel data statistics can be added according to the actual needs of urban public transportation development, so as to adapt to the new development needs of urban bus informatization.

The system has built a perfect security protection mechanism. For the management end, strict identity authentication process is enforced to prevent illegal access; sensitive information such as administrator passwords is stored by encryption technology to eliminate the possibility of data leakage from the technical level [19]. For the user end, the legality of user requests is verified to prevent SQL injection and other malicious attacks; user personal information is encrypted and stored to ensure the confidentiality and integrity of user data. In addition, the system records the key operation logs of users, which is convenient for security audit and fault investigation.

The interface design of the system is simple and beautiful with a reasonable layout and clear operation process, which reduces the user's operation cost[20]. The system provides friendly error prompts and operation guidance, and users can quickly master the use method of the system without professional training. At the same time, the system runs stably 24/7 with short fault recovery time to ensure that users can query bus information at any time.

## **4. System Design**

Based on the demand analysis, the system is divided into two major modules: the management end and the user end, and each module is subdivided into specific functional sub-modules. The management end includes user management, line management, train number management, bus line management, bus station management, bus information management, surrounding service management, bus dynamic management and system management sub-modules; the user end includes bus line query, bus station query, bus information query, surrounding service query and announcement information browsing sub-modules. The modular design makes the system function clear, the code structure simple, and convenient for the later development, maintenance and function expansion of the system.

### **4.1 Detailed Design**

The management end is the core of system data management, and the detailed design of each sub-module is carried out according to the functional requirements, and the corresponding operation flow is formulated to ensure the standardization and efficiency of the administrator's operation. The

user management process: the administrator can add new users by clicking the add button, search for existing users by user account or name, and modify or delete the queried user information; if the queried user does not exist in the database, the system prompts a query failure. The line management process: the administrator submits a line add, delete, modify and query request through the background interface, the system automatically verifies the data legality (such as line duplication) and executes the database operation, and returns the operation result (success/failure) to the interface in real time. The bus line management process: the administrator can operate the bus line by clicking the add, delete, modify and query buttons, query the bus line by entering the line name, starting station, terminal station or transfer information, the system automatically verifies the line duplication, saves the data, and pops up the operation result in real time; at the same time, the administrator can view the user's comments on the line and reply in time. The design of other sub-modules such as train number management, bus station management and bus dynamic management follows the above design ideas, with clear operation processes and strict data verification to ensure the accuracy of system data.

The user end is oriented to ordinary users, and the detailed design is based on the principle of user-friendliness, simplifying the operation process and improving the query efficiency. The bus line query process: the user clicks the bus line to enter the query interface, enters the starting station, terminal station or transfer information to query the relevant line information; if the queried bus line does not exist in the database, the system prompts an operation failure, and the user can also evaluate, like or collect the line. The bus station query process: the user clicks the bus station to enter the query interface, enters the station name, road name or station location to query the relevant station information, and clicks the station name to view the detailed information such as buses passing through the station and surrounding stations. The bus information query process: the user enters the bus number, shift, starting station or terminal station in the search box to query the relevant bus operation information, and the query results are displayed in the form of a list for the user to view. The surrounding service query and announcement information browsing processes are designed to be simple and intuitive, and users can quickly query the required information by entering keywords, which improves the user experience.

## 4.2 Database Design

Database conceptual design carries out abstract planning around the data stored in the database and its relevance, focusing on analyzing the logical relationship, attribute characteristics and constraint conditions of data entities without involving the physical structure of specific data tables [21]. The system selects the MySQL database to realize the data storage function, and completes the standardized management and persistent storage of core information such as users, bus lines and bus stations through the scientific design of the data table structure. The main entities of the system include user, line, train number, bus line, bus station, bus information, surrounding service, bus dynamic and system, and there are certain association relationships between each entity. For example, a bus line corresponds to multiple bus stations, and a bus station corresponds to multiple bus lines; a bus information entity is associated with the bus line entity and the train number entity. The E-R diagram of the system clearly reflects the attribute characteristics of each entity and the association relationship between entities, providing a theoretical basis for the subsequent design of the database table structure.

According to the database conceptual design and the functional requirements of the system, the key data tables of the MySQL database are designed, including user table, line table, train number table, bus line table, bus station table, bus information table, surrounding service table, bus dynamic table, about table, announcement table and announcement classification table [22]. The user table (user) stores user information including user name, password, avatar and role; the line table (xianlu) stores line name and addition time; the train number table (checi) stores train number name and addition time; the bus line table (gongjiaoxianlu) is the core table of the system, storing detailed information of bus lines such as bus number, train number, starting station, terminal station, passing stations, fare and operation time; the bus station table (gongjiaozhandian) stores station information such as station name, bus line, road name, station location and arrival time; the bus information table (gongjiaoxinxi)

stores bus operation information such as bus number, shift, first and last bus times and departure interval; the surrounding service table (zhoubianfuwu) stores the surrounding service information of bus stations such as title, service type and detailed description; the bus dynamic table (gongjiaodongtai) stores the real-time dynamic information of bus stations such as passenger flow and congestion; the about table stores the basic information of the system such as the introduction of the system and the information of the developers; the announcement table (news) and the announcement classification table (newstype) store the announcement information and its classification released by the system. All data tables take the id field as the primary key, and set reasonable field types and lengths to ensure the integrity and consistency of data; at the same time, indexes are created for the frequently queried fields to improve the query efficiency of the database.

## 5. System Implementation

### 5.1 Management End Implementation

The administrator enters the management end page and clicks user management to enter the user management page. The administrator can add new users by clicking the add button and entering the new user information; the system verifies the uniqueness of the user name, and if the user name already exists, it prompts an operation failure. The administrator can also query the existing users in the database by entering the user account or name in the search box, and the queried users are displayed in the list; the administrator can view the detailed information of the user, and perform modification and deletion operations. The core code of user management mainly includes the save method for adding users and the update method for modifying users, which realizes the verification of user name uniqueness and the addition and modification of user data through the MyBatis framework. Line management: the administrator clicks line management to enter the line management page, adds new line information by clicking the add button, and the system verifies the uniqueness of the line name; the administrator can query the existing line information in the database by entering the line name in the search box, and perform view, modify and delete operations on the queried lines. The core code includes the update method for modifying lines, which realizes the full update of line data. Train number management: the implementation process is similar to line management, the administrator can add, query, modify and delete train number information, and the core code includes the query method for querying train numbers, which realizes the precise query of train number data through the EntityWrapper.

Bus line management: the administrator clicks bus line management to enter the management interface, adds new bus line information by clicking the add button, and uploads the line picture; the system verifies the format of the uploaded picture, and if the format is incorrect, it prompts to re-upload. The administrator can query the relevant bus line information by entering the line name, starting station, terminal station or transfer information, and perform modify and delete operations on the queried lines; at the same time, the administrator can view the number of bus lines and train numbers in the form of pie charts and bar charts, and view and reply to user comments. The core code includes the update method for modifying bus lines, which realizes the full update of bus line data. Bus station management: the administrator adds new bus station information by clicking the add button and uploading the station picture; the system verifies the picture format and the uniqueness of the station name. The administrator can query the relevant station information by entering the station name, bus line, road name or station location, and perform view, modify and delete operations. The core code includes the update method for modifying bus stations, which realizes the full update of station data.

Bus information management: the administrator can add new bus information by entering the bus number, shift, starting station and other information, and query the relevant bus information by multiple conditions; the core code includes the save method for adding bus information, which realizes the insertion of bus information data. Surrounding service management: the administrator adds new surrounding service information by entering the title and service type, and queries the relevant service information by the title; the core code includes the delete method for deleting surrounding services, which realizes the batch deletion of service data. Bus dynamic management:

the administrator adds new bus dynamic information by entering the bus station, passenger flow and other information, and queries the relevant dynamic information by the bus station; the core code includes the update method for modifying bus dynamics, which realizes the full update of dynamic data. System management: including the management of about us, system introduction, carousel map and announcement information; the administrator can modify the basic information of the system, upload the carousel map picture, and add, modify and delete announcement information; the core code includes the save method for adding system information, which realizes the insertion of system data.

## 5.2 User End Implementation

Bus line query: the user clicks the bus line on the user end homepage to enter the query interface, and can query the relevant line information by entering the starting station, terminal station or transfer information; the query results are displayed in the form of a list, and the user can click the line name to view the detailed information such as passing stations, first and last bus times and fares, and can also evaluate, like or collect the line. The core code includes the query method for querying bus lines, which realizes the multi-condition query of line data through the EntityWrapper. Bus station query: the user clicks the bus station to enter the query interface, and queries the relevant station information by entering the station name, road name or station location; the user can click the station name to view the detailed information such as buses passing through the station, arrival time and surrounding stations. The core code includes the query method for querying bus stations, which realizes the multi-condition query of station data.

Bus information query: the user clicks the bus information to enter the query interface, and queries the relevant bus operation information by entering the bus number, shift, starting station or terminal station in the search box; the query results are displayed in a list, and the user can click the bus number to view the detailed information such as passing stations, first and last bus times and departure interval. The core code includes the query method for querying bus information, which realizes the multi-condition query of bus information data. Surrounding service query: the user clicks the surrounding service to enter the query interface, and queries the relevant service information by entering the service title; the user can click the service picture to view the detailed description of the surrounding service, which provides a supporting reference for the user's travel. The core code includes the query method for querying surrounding services, which realizes the query of service data by title.

The user clicks the announcement information to enter the query interface, and can query the relevant announcement information by entering the announcement title; the announcement information is displayed in the form of a list, including the announcement title, release time and brief introduction. The user can click the announcement title to view the detailed content of the announcement, and can also like and collect the announcement. The core code includes the query method for querying announcement information, which realizes the query of announcement data by title, and improves the user's browsing experience.

## Acknowledgements

This research is supported by the teaching research project of Nanyang Institute of Technology. The authors would like to express their sincere gratitude to Professor Chao Wang for his valuable guidance and selfless help in the whole process of the graduation design. His professional knowledge and rich teaching experience have provided important support for the completion of this research. The authors also thank the family and friends for their constant support and encouragement, and the classmates for their help in the development and testing of the system. In addition, the authors thank the references cited in this paper and the anonymous reviewers for their valuable comments and suggestions, which have greatly improved the academic level of this paper.

## References

- [1] X.H. Wang, F.F. Zhang and G. Wu: Research on the Development and Application of Intelligent Bus System, *Journal of Public Transport*, Vol. 14 (2024), p.139-141. (In Chinese)
- [2] R. Hu and X.L. Yang: Design and Implementation of Web System with Separate Front and Back Ends Based on Spring Boot, *Journal of New Energy and Intelligent Connected Vehicles*, Vol. 01 (2024), p.88-97. (In Chinese)
- [3] S. Luo and J. Yang: Research on the Optimization of Intelligent Bus Dispatching System and Its Application in Modern Urban Transportation, *Journal of Transportation*, Vol. 03 (2025), p.66-69. (In Chinese)
- [4] W.N. Zhang and J.R. Wang: Analysis of the Development of Urban Public Transportation under the Background of Transportation Informatization, *China Collective Economy*, Vol. 20 (2024), p.8-11. (In Chinese)
- [5] Y. Liu and R.X. Li: Building a "Bus City" and Constructing a Happy Tonghua, *Tonghua Daily*, 2022-09-28(001). (In Chinese)
- [6] X.W. Li: Analysis of Vue.js Front-end Application Technology, *Network Security Technology and Application*, Vol. 04 (2022), p.44-45. (In Chinese)
- [7] Z.Q. Lv: Design of Management System Based on Vue Front-end Development Framework, *Computer Knowledge and Technology*, Vol. 36 (2024), p.56-59. (In Chinese)
- [8] Q. Kong: Design and Implementation of Current Periodical Query System in Reading Room Based on Spring Boot+Vue Framework, *Modern Information Technology*, Vol. 07 (2025), p.98-102+108. (In Chinese)
- [9] S.Y. Kuai, J.X. Li and M. Wang: Design and Implementation of Tutor-graduate Dual Selection System Based on Spring Boot+Vue, *Modern Information Technology*, Vol. 06 (2025), p.20-25+32. (In Chinese)
- [10] L.Y. Chen: Design of Self-designed Course Online Examination Management System Based on Spring Boot, *China New Information*, Vol. 06 (2025), p.83-85. (In Chinese)
- [11] H.J. Ouyang, M. Ge and H.B. Cheng: Research on the Application of MyBatis Framework in Data Persistence Layer, *Microcomputer Applications*, Vol. 01 (2023), p.73-75. (In Chinese)
- [12] Z.L. Wang and S.B. Ji: Interface Design between Web Front-end and Database Based on SpringBoot, *Industrial Control Computer*, Vol. 03 (2023), p.51-53. (In Chinese)
- [13] J.L. Xiang: Research on the Application of MySQL Database Technology in Campus Information Management, *Information Recording Materials*, Vol. 03 (2025), p.104-106. (In Chinese)
- [14] Z.T. Wang: Development and Design of Project Management Information System Based on B/S Mode, *Office Automation*, Vol. 24 (2024), p.84-86. (In Chinese)
- [15] C.M. Li, H.C. Zhang and Y.L. Liang: Design and Implementation of Work-study Information Management System Based on Web B/S Architecture, *Technology Innovation and Application*, Vol. 27 (2024), p.131-134. (In Chinese)
- [16] Z.H. Tuo: Comparative Study Based on Black-box Testing and White-box Testing, *Electronic World*, Vol. 11 (2021), p.55-56. (In Chinese)
- [17] Y. Zha: Strategies and Methods of Software Testing, *Information and Computer (Theory Edition)*, Vol. 10 (2021), p.67-69. (In Chinese)
- [18] M.L. Zhang, J.C. Lv and Y.B. Jiang: Intelligent business cloud service platform based on SpringBoot framework, in: *Proc. Asia-Pacific Conference on Image Processing, Electronics and Computers*, Dalian, China, 2020, p.201-207.

- [19] Y. Quan: Design and Implementation of E-commerce Platform based on Vue.js and MySQL, in: Proc. International Conference on Computer Engineering, Information Science & Application Technology, Chongqing, China, 2019, p.449-454.
- [20] J. van der Geer, J.A.J. Hanraads and R.A. Lupton: The Art of Writing a Scientific Article, Journal of Scientific Communication, Vol. 163 (2000), p.51-59.
- [21] W. Strunk Jr. and E.B. White: The Elements of Style, Third ed., Macmillan, New York, 1979.
- [22] G.R. Mettam and L.B. Adams: How to Prepare an Electronic Version of Your Article, in: B.S. Jones, R.Z. Smith (Eds.), Introduction to the Electronic Age, E-Publishing Inc., New York, 1999, p.281-304.
- [23] R.J. Ong, J.T. Dawley and P.G. Clem: Submitted to Journal of Materials Research, Vol. (2003).
- [24] P.G. Clem, M. Rodriguez and J.A. Voigt: U.S. Patent 6,231,666, (2001).
- [25] Information on <http://www.weld.labs.gov.cn>
- [26] C.H. Xu: Design, Simulation and Application of Composite Ceramic Tool Materials, (Ph.D., Shandong University of Technology, China 1998), p.28.
- [27] G. Bräuninger: Proc. International Workshop on Diamond Tool Production, (Turin, Italy, November8-10,1999),Vol.1,p.154