A Dynamic Model for Web-based Learning System Design

JIHONG CHEN[1] WANWU GUO[2]

[1] Chaoyang Library, Jilin University
6 Ximinzhu Street, Changchun, Jilin Province, China

[2] School of Computer and Information Science
Edith Cowan University
2 Bradford Street, Mount Lawley, WA 6050, AUSTRALIA

Abstract - Web-based learning systems have become more and more popular for universities to support their teaching and learning activities. Such a system normally requires progressive development under extreme time constraint in relation to the academic calendar and in the presence of some dissatisfaction known prior to the development. At any stage of the development, new service that was not included in the current phase of development but with urgent demand by users may be required to be added into the system. Developing such a system needs a dynamic design model, instead of any existing 'static' approaches. In this paper we propose a dynamic model based on prioritisation of characterisation of distributed systems for the design of Web-based systems. Application of this model in development of the Information Retrieval Learning System Online (IRLSO) at Jilin University in China shows that this model is useful in not only dynamically guiding the development of a Web-based system, but also prioritising tasks for the successive development.

Key-Words: - Web-based learning systems, Dynamic design model, Characterisation of distributed systems, Information Retrieval Learning System Online (IRLSO)

1 Introduction

Nowadays more and more universities have been moving towards using Web-based systems to support their teaching and learning activities. A Web-based system provides students with high accessibility in getting study resources, portability in managing teaching materials, and more flexibility in organising their own schedule under the semester framework. It also provides instructors with a high degree of maintainability and reusability in course updates and redevelopment. However, development of such a Web-based system and consequent updating are always under extreme time constraint in relation to the academic calendar. Thus any Web-based system is a result of successive development of a multi-phased project. Updating the existing system and making new development to accommodate a growing number of demands in various aspects are also needed during the course. At any stage of the development, some new services that were not included in the development of current phase but with urgent demand by users may be required to be added into the system. This brings a question - how to design the development of such a system which is dynamic and known to be prone to some dissatisfaction. There could be no suitable answers to this question if we use existing 'static' approaches (e.g., Sommerville, 2001; Pressmanl, 2001). We need a dynamic design model to deal with this situation.

We may have many ways to design such a system based on different models. In this paper we regard a Web-based system as a distributed system that provides document-based services to users located in different places. Thus we can use characterisation of distributed systems to guide the development of any Web-based systems. We firstly analyse characterisation of distributed systems, and then propose a dynamic model based on the prioritisation of the characterisation of distributed systems for designing any Web-based systems. This model is then used to analyse the development of the Information Retrieval Learning System Online (IRLSO) that has been used since 2002 at Jilin University in China.

2 Characteristics of Distributed Systems

Tanenbaum and van Steen (2002) state that any distributed systems should be able to achieve shareability, transparency, openness, and scalability. Coulouris et al (2001) have the same viewpoints on these four aspects, and furthermore add that distributed systems should be able to deal with heterogeneity, security, failure handling, and concurrency, with which Tanenbaum and van Steen (2002) also agree but classify into technical layers. These features of distributed systems are summarised below.

Shareability is the ability that the comprising entities can use each other's resources. The sharing of resources (hardware, software and information) is the main motivation for constructing distributed systems.

Concurrency in a distributed system is the ability to process multiple tasks at the same time.

Security refers to the fact that a distributed system should not only allow its users to share resources more easily, but also protect these resources against any unauthorized activities with them.

Failure handling is the feature that failures in a distributed system should be partial, i.e., some components fail while others continue to function.

Heterogeneity is referred to the fact that a distributed system should allow users to access services and run applications over a heterogeneous collection of computers and networks. Heterogeneity applies to networks, computer hardware, operating systems, programming languages, and implementations by different developers.

Openness is that a distributed system should allow the system to be extended and re-implemented in various ways. It is measured primarily by the degree to which new resource-sharing services can be added without disruption or duplication of existing services, and be made available for use by a variety of client programs.

Scalability in a distributed system is the characteristic that a system remains effective when there is a significant increase in the number of resources and the number of users.

Transparency is defined as the concealment of the separation of components in a distributed system so that the system is able to present itself to users and applications as if it were only a single computer system.

In fact, these features are relevant to different scopes in constructing a distributed system. Shareability is the main goal for constructing distributed systems. The others are system design requirements to achieve the shareability in a certain degree of satisfaction. Thus, in theory these features are ideal desires for any distributed systems to have, but in practice there are only relative criteria for building up a specific distributed system in a certain degree of satisfaction in all or some of the requirements.

3 A Dynamic Model for Web-based System Design

Now that *Shareability* is the main goal for constructing distributed systems and other features are design requirements to achieve the shareability in a certain degree of satisfaction, these features can be prioritised into different classes in relation to the necessity to achieve the shareability. There are surely many ways to classify these features. In this paper we classify these features in distributed systems into purposive, utility, extensible, and accessory classes.

The fundamental features of a system can only be included in the purposive class. Obviously only shareability realised with basic functionality to a system can fit into this class. Any features necessary to make such a system functional in a certain degree of satisfaction should be classified into the utility class. These include concurrency, failure handling, and security. Concurrency in a distributed system ensures that the system can handle synchronous requests and provide multiple services to its users at the same time, which is vital to achieve shareability in a distributed environment. Failure handling makes the system recoverable once a failure occurs, which protects the system from disastrous loss in shared resources. Security provides necessary protection on shared resources from illegal activities on them and thus gives users trust in using the system. Therefore, purposive and utility classes together form the core of a system and provide fundamental operations for a distributed system to work reliably at least in a homogeneously distributed environment.

With this workable system, any measures related to promoting this system to heterogeneous environments, and improving flexibility and extensibility of this system should be classified into the extensible class, these including heterogeneity, openness, and scalability. It is clear that extensible class is built on top of the system core or the combination of purposive and utility classes. Any features that can be relaxed without degrading the utilisation of a system should be classified into the accessory class, this being the case of transparency and any other additions. This classification is summarised in Table 1.

When we take into account the fact that a Web-based system is a result of successive development of a multi-phased project, this classification can then be modified as a target-oriented model shown in Figure 1, in which the targets are set to all the classes in different phases of the development. The target boundaries are dynamic, which indicates the flexibility in adding new targets for further development. The current boundary positions represent the system requirements that should be met in the completion of the current development phase. It is clear that this is a dynamic and target-oriented design model.

Table 1: Classification for the major features of distributed systems

Class	***Priority***	***Feature***
Purposive	1	Shareability through basic functionality
Utility	2	Concurrency, Failure handling, Security
Extensible	3	Heterogeneity, Openness, Scalability
Accessory	4	Transparency, Others

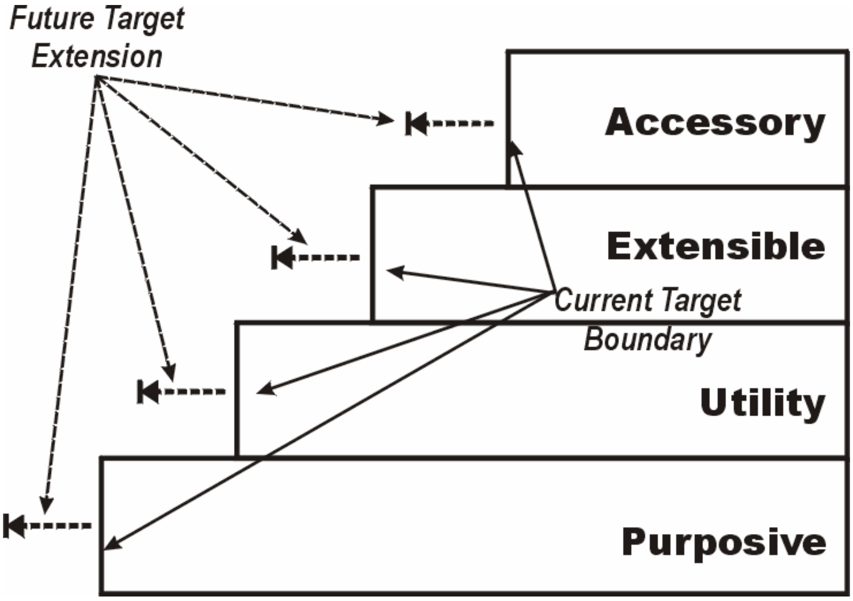

Figure 1 Structure of the dynamic design model

4 Analysis of Development of Information Retrieval Learning System Online (IRLSO)

IRLSO is a Web-based teaching and leaning system that provides support for students leaning skills in information searching and referencing. The primary objective of the initial development is to produce a Web-based system that could provide both students with remote access to teaching materials in electronic format and instructors with basic capability in adding, removing, and updating teaching materials. The main interface of this system is shown in Figure 2. This system consists of three parts: Learning IR Online, Messaging Online, and Testing Online (Figure 3).

Learning IR Online is the core of this system. It provides all the required teaching materials organised as modules consistent with weekly teaching schedule. Links to some national and international institutions/organisations that provide information retrieval services are also given in this part to support student self-learning. This part was developed first to form the foundation of this system for both the initial use for teaching and learning IR course online, and further system development. This actually realised the primary tasks of the purposive and utility classes for this IR system.

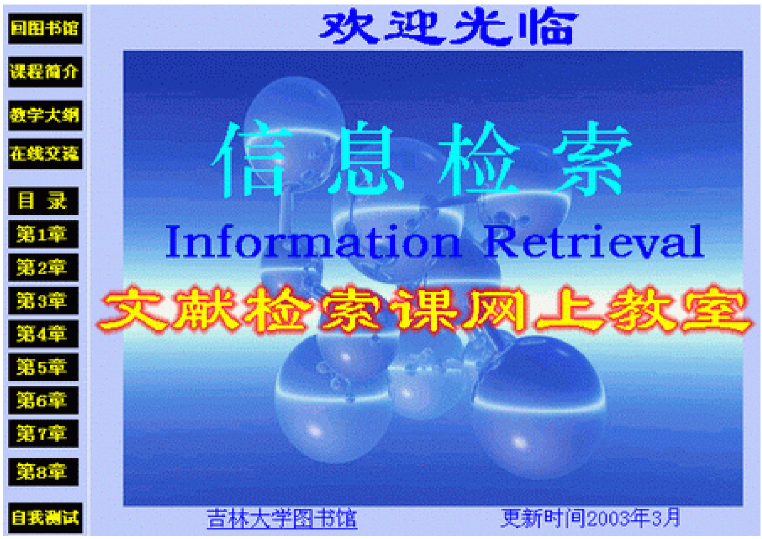

Figure 2 Main interface of IRLSO

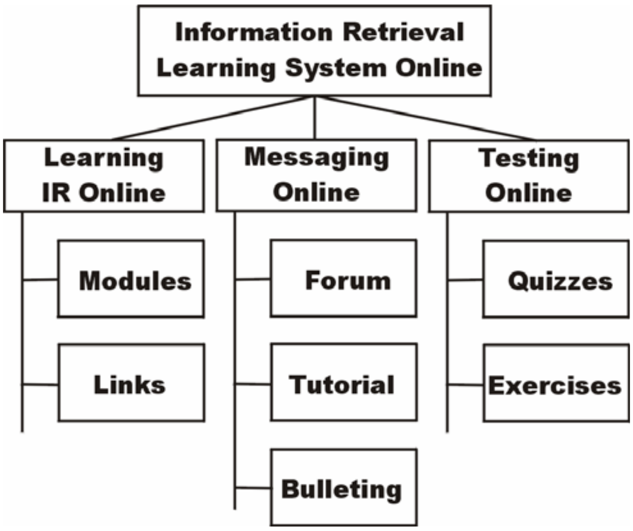

Figure 3 Structure of IRLSO

Messaging Online is then added into the initial version of the system to support online tutorials and forum. The online tutorial service is an addition to the purposive class whereas the forum belongs to the extensible class. Bulleting is also provided to the instructor in this part to remind students of any updates made to the existing teaching materials and any important events approaching. This can be regarded as the addition to the accessory class because the same message can also be passed onto students in the Learning IR Online part by inserting the message into the corresponding sessions. However, Bulleting would be more popular to students.

Testing Online is another addition to the utility class to enhance the system’s functionality. In this part students can do self-assessment using online quizzes and/or exercises. After these two major updates, this system now becomes more powerful compared with its initial configuration. However, more work still needs to do to promote this system working with other platforms in the future, not only with the Windows system. This should be the major task for the next phase of system development, which belongs to the expansion of the extensible class.

5 Conclusion

Although IRLSO is still under further development, according to our dynamic design model, the whole system development and each individual development phase have been well managed. This target-oriented model allows the presence of partial exclusion of some services to the system in different development phases. The initial version of a system should make the core, or combination of purposive and utility classes, functional reliably; then other improvements can be successively built on this foundation. Given the nature of any Web-based system being a result of successive development under extreme time constraint in relation to the academic calendar, this dynamic and target-oriented model is useful in not only guiding the development of such a system, but also prioritising targets for its successive development.

References

[1] I. Sommerville, *Software engineering*, Addison-Wesley, 2001.

[2] R. Pressman, *Software engineering: a practitioner's approach*, McGraw-Hill, 2001.

[3] A. Tanenbaum, M. van Steen, *Distributed systems: principles and paradigms*, New Jersey: Prentice Hall, 2002.

[4] G. Coulouris, J. Dollimore, T. Kindberg, *Distributed systems: concepts and design*, Addison-Wesley, 2001.