Research on Web2.0 System Design Based on CAS Theory

Kai Chen and Hengshan Wang

University of Shanghai for Science and Technology,
Shanghai 200093, P.R.China
kookia.chen@gmail.com, wanghs@usst.edu.cn

Abstract. According to complexity theory, this paper analyses several characteristics of some present Web2.0 systems, such as Blog, Wiki, SNS and social tags. It also summarizes the disadvantages of current information system design methods and finally re-designs it based on CAS and DSDM.

Keywords: Complex adaptive systems (CAS) Web2.0.

1 Introduction

The concept of "Web 2.0" began with a conference brainstorming session between O'Reilly and MediaLive International. Dale Dougherty, web pioneer and O'Reilly VP, noted that far from having "crashed", the web was more important than ever, with exciting new applications and sites popping up with surprising regularity [1]. Though there is no common answer to what is Web 2.0, in the next few years the core ideas unknowingly and continued to infiltrate into the Internet. Users gradually changed from the recipient of information into the producers and disseminators, from the audience to their own social group who have the right to speak, and gradually shift from an individual to a social group with a common concern, the Internet services model also gradually change.

With the idea of Web2.0 widely understanding, and information system evolves from low-level to advanced, simple to complex, close to open, and isolated to cooperative, The components of information system become more dependent, the coupling degree among components gets lower, while the components interact and collaborate more flexibly. In a word, information system shows more characters of complex system than before. The most prominent ones are web2.0 systems. Therefore, it becomes a growing real need that using some theories and methods of related with system complexity research to study and design the information system.

2 CAS Theory

2.1 The Core of CAS

The core of CAS theory is that adaptation makes complexity [2]. Complex adaptive systems are special cases of complex systems. They are complex in that they are

J. Zhou (Ed.): Complex 2009, Part I, LNICST 4, pp. 188 – 195, 2009.

diverse and made up of multiple interconnected elements and adaptive in that they have the capacity to change and learn from experience [3]. A Complex Adaptive System (CAS) is a dynamic network of many agents (which may represent cells, species, individuals, firms, nations) acting in parallel, constantly acting and reacting to what the other agents are doing. The control of a CAS tends to be highly dispersed and decentralized. If there is to be any coherent behavior in the system, it has to arise from competition and cooperation among the agents themselves. The overall behavior of the system is the result of a huge number of decisions made every moment by many individual agents. The term complex adaptive system (CAS) was coined at the interdisciplinary Santa Fe Institute (SFI), by John H. Holland, Murray Gell-Mann and others.

2.2 General Properties

Despite substantial different systems center on them in detail, CAS share four major features [4]:

1. Parallelism. CAS consists of large numbers of agents that interact by sending and receiving signals. Moreover, the agents interact simultaneously, producing large numbers of simultaneous signals.
2. Conditional action. The actions of agents in a CAS usually depend on the signals they receive. That is, the agents have an IF/THEN structure: IF [signal vector x is present] THEN [execute act y]. The act may itself be a signal, allowing quite complicated feedbacks, or the act may be an overt action in the agent's environment.

Interlocking sequences of signal-processing rules become programs that are executed in parallel, with all that implies for flexibility and breadth of repertoire.

3. Modularity. In an agent, groups of rules often combine to act as "subroutines". For example, the agent can react to the current situation by executing a sequence of rules. These "subroutines" act as building blocks that can be combined to handle novel situations, rather than trying to anticipate each possible situation with a distinct rule. Because potentially useful building blocks are tested frequently, in a wide range of situations, their usefulness is rapidly confirmed or disconfirmed.
4. Adaptation and evolution. The agents in a CAS change over time. These changes are usually adaptations that improve performance, rather than random variations.

3 Research on Present Characteristic Web2.0 Systems

In recent years, thousands of new web information systems appeared on the internet which based on the conception of Web2.0. Blog, Wiki, Social Network Service, and Social Tags are the four main kinds of Social Software [5] which are recognized as applications of Web 2.0. From the view of CAS (Complex Adaptive System) theory, Web 2.0 system has characteristics of aggregation, nonlinearity, flows and diversity, mechanisms of tagging, internal model and building blocks [6]. These characteristics and mechanisms are recognized by Holland [2] as the seven basic points of CAS.

Different from traditional software and information system, web2.0 systems have some typical characters as follows:

1. They use more participative-architecture and open-architecture; the most important is that people become a part of design of system;

2. There are plenty of nonlinear and self-organized mechanisms in system, these mechanisms can make the system adjust its function and structure continually to adapt to the changing environment;

3. The interactions and inter-operations among these systems become very frequent and complex than ordinary ones, the complex relation of the interactions and inter-operations facilitate a dynamic complex network coming into being, The complex network, which is quite similar to ecology network;

4. Social network analysis has been applied to information system design as embedded algorithms. In these systems, spontaneous cooperation among users allows many kinds of social networks to grow up from the bottom.

According to observation, user' enthusiasm of creativity started to descend, and the interaction among participators is increasing. How can we find the change when we design an information system? This problem has been raised in the paper [4]. So in the new method of information system design will be concerned, one is that keep the enthusiasm of participators, the other is that how to find the rule of system's adaptation and evolution. And we never deem these types of web2.0 systems as isolated ones, they are a whole.

4 The New Method of Web2.0 Information System Design

4.1 The Trend of Information System Design and Development

With the rapid development of information technology, the design and development of information systems also took place change time and again. From the evolution and development patterns of information systems, they were the earliest single-user, stand-alone system (or a host with multiple terminals), then developed the client/server model (C/S) systems which were used in the local area network or a closed network environment, after that, browser/server (B/S) model systems were used in the open environment of the Internet, as well as ,the expansion of the three-tier and multi-layer structure based on B/S model. Recently, developed to the so-called fat client model and decentralized peer-to-peer (P2P) model, information systems were toward more personalized, intelligent, distributed and decentralized. We can see from the Figure 1.

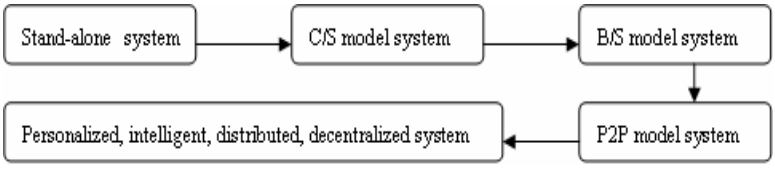

Fig. 1. The evolution and development of information system

4.2 Present Method of Development and Design

In the software engineering domain, software or information system development process is a structure imposed on the development of a software product. Synonyms include software life cycle and software process. There are several models for such processes, each describing approaches to a variety of tasks or activities that take place during the process.

The best-known and oldest process is the waterfall model, where developers are to follow these steps in order: Requirements specification, Design, Construction (implementation or coding), Integration, Testing and debugging (validation), Installation (deployment), Maintenance. After each step is finished, the process proceeds to the next step, just as builders don't revise the foundation of a house after the framing has been erected.

Iterative development is a cyclic software development process developed in response to the weaknesses of the waterfall model. The basic idea behind iterative enhancement is to develop a software system incrementally, allowing the developer to take advantage of what was being learned during the development of earlier, incremental, deliverable versions of the system. Learning comes from both the development and use of the system, where possible. Key steps in the process were to start with a simple implementation of a subset of the software requirements and iteratively enhance the evolving sequence of versions until the full system is implemented. At each iteration, design modifications are made and new functional capabilities are added.

The traditional approaches of system design have a common feature, it's top-down, according to needs design and analysis system. All of the design, structural order and classification are pre-determined by the designer, once the demand changing, it is difficult to make rapid and adaptive responding. Designer doesn't take into account that users' involvement is a key part of information system design. To methodology, they are a product of dualism, the relationship of user and system is use and used, with the change of requirements, traditional approaches are difficult to adapt themselves.

Agile development and dynamic systems development method (DSDM) bring us a new hope. DSDM is one of a number of agile methods for developing software. It is an iterative and incremental approach that emphasizes continuous user involvement. Its goal is to deliver software systems on time and on budget while adjusting for changing requirements along the development process. There are 9 underlying principles consisting of four foundations and five starting-points [7].

4.3 The Method of Web2.0 Information System Design

In a paper called "The Dynamic Business Applications Imperative," John R. Rymer, a senior analyst with Forrester, points to a fundamental shortcoming of today's applications: Today's applications force people to figure out how to map isolated pools of information and functions to their tasks and processes, and they force IT pros to spend too much budget to keep up with evolving markets, policies, regulations, and business models. IT's primary goal during the next five years should be to invent a new generation of enterprise software that adapts to the business and its work and evolves with it. Forrester calls this new generation Dynamic Business Applications, emphasizing

close alignment with business processes and work (design for people) and adaptability to business change (build for change).

In the book of "The Fifth Discipline", Peter Senge said: From a very early age, we are taught to break apart problems, to fragment the world. This apparently makes complex tasks and subjects more manageable, but we pay a hidden, enormous price. We can no longer see the consequences of our actions; we lose our intrinsic sense of connection to a larger whole. When we then try to "see the big picture," we try to reassemble the fragments in our minds, to list and organize all the pieces. But, as physicist David Bohm says, the task is futile—similar to trying to reassemble the fragments of a broken mirror to see a true reflection. Thus, after a while we give up trying to see the whole altogether [8].

After analyzing the features of present Web2.0 system, the trend of information system and the characters of present approaches of development, now we will put these fragments together, design a method of Web2.0 information system development combined with CAS theory and DSDM. Figure 2.

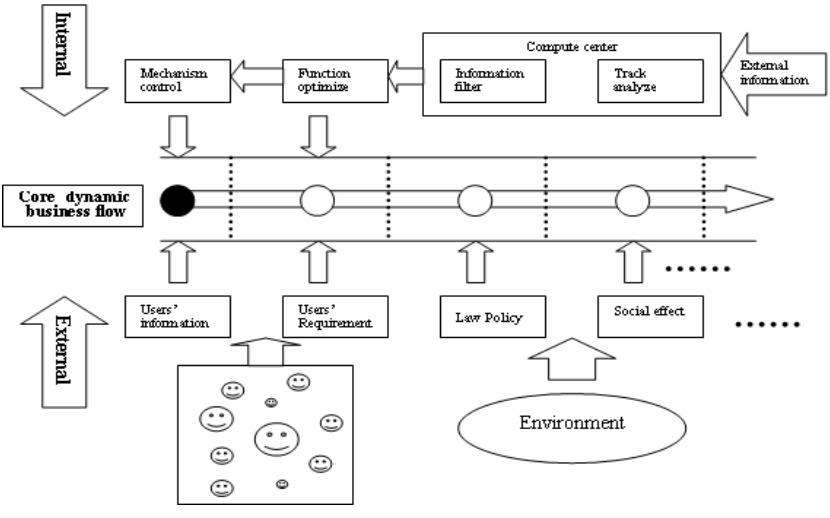

Fig. 2. The method of Web2.0 information system design

1. Core dynamic business flow

According to different circumstances, core business, as an engine of system, must be designed the fundamental business and service. Allow users to participate in the entire process of dynamic business and applications, in the process of participation exert their creativity, encourage them to cooperate with their friends or partners. With incentives make their participation more meaningful, and share the joy of success. In this way user will maintain long-term enthusiasm for participation. And unlike most of the SNS is made only for entertainment with your partners. To reach adaptability of system's business flow, every state is open, and can accept information from internal

control and external feedback. Own the adaptability, and system can supply better services and applications.

2. External Feedback

Based on the theory of CAS, the constitution of a complex adaptive system has agent and environment. In the approach of web2.0 information system development and design, people serve as an important part; they are not only the creators of system's information, but also the foundation of the self-organization and creators of emergence. So user involvement is a prominent character. So-called environment is laws, policies and social effects and others. The information will be collected and transmitted to compute center of internal control for system's improvement. At the same time, the maintainer will realize dynamic reflection of user's requirements and environment's change, in order to optimize the program in compute center.

3. Internal control

Just agents and environment, we can not get our goal. We need another one, it is mechanism. What is mechanism? It is the rule of user's behavior in the system, there is no system can run without regulation. Agent, environment and regulation can create complexity, and then the system can evolve. We don't know where the system will go, but we can control the regulation of the system, via these mechanism lead and help the evolution of system.

Compute center. Its main work is to find the rule of system evolution. The rule is not permanent, and can change with the external feedback information. Filtering information, analyze and track record of users' behavior is the specific ways. Creators' information and behavior record are collected by compute center, and then through the calculation of algorithm and statistic analysis get the related data. At this time, maintainer of the system will give a report about suggestion of optimizing function and the situation of users' behavior and interests based on external environment and related data. Compute center will be optimized by programmer accordance with the report. In order to realize better individual service, we record users' behavior. With constantly improved, users also feel more in line with individual needs and preferences.

Functions optimize. A regulation is a function component, it is needed to realize by programmer. The main work of this part is to manage, realize and optimize these components based on report of compute center. In order to realize the adaptability of the information system, it supplies the interface for the application components which used in other systems. Of course, if you are professional user, the system would accept your own application through API. Anther important work is to optimize the core dynamic business flow.

Mechanism control. According to real requirement, design, add or delete the regulation of information system. If the regulation is good for the evolution or system, we will add it, or not.

Through users' participation, external feedback, dynamic control and dynamic management, new generation information system has its life. The more run, the better will be. With the user involvement, information system has the complexity. As designer, only can provide a platform for services and applications, as to how the system develop, we can not predict. If only an information system has the characters of complex adaptive system, it has its own evolution; only can we guide its development

through the mechanism. So we must do a rapid response when requirement changed from users' involvement information, and gradually improve the mechanism to guide the development of the system's evolution. As a single user in the system, even if you asked him what good or not, what does not need or not, it is very difficult to get an unequivocal answer, because they are not sure. The solution of this problem depends on the analysis of the external feedback information. To realize this process, on the one hand, directly listen to the suggestions of users, on the other hand, complete through statistical analysis and psychological analysis of users' behavior, it has a big different with traditional design of information system which only focuses on technology. So at the beginning of design, system will be designed the tables to record users' behavior data. After analyzing these data, one is that designer can understand the needs of individual users, provide better personalized service in the future; the other is that can know group behaviors, improve the quality of service in system.

5 Conclusions

In this paper, analyzing the characters of present Web2.0 system, the trend of information system development and the situation of traditional design approaches, then build a new method of Web2.0 information system development and design based on CAS theory and DSDM. In order to realize the convenient management, the whole design is divided into three parts, including core dynamic business flow, external feedback and internal control, which are dynamically controlled and managed by users' participation and information feedback to make the system work effectively. The system may like getting a real life and having an evolution process and finally realize the adaptation for users' personality. That's the Web2.0 adaptive information system we want to get.

Acknowledgement

This paper is supported by Shanghai Key project Management Science and Engineering under Grant Number S30504.

References

1. O'Reilly, T.: What Is Web 2.0: Design Patterns and Business Models for the Next Generation of Software. O'Reilly Media, Inc., Sebastopol (2007), http://www.oreillynet.com/pub/a/oreilly/tim/news/2005/09/30/what-is-web-20.html
2. Holland, J.H.: Hidden Order: How Adaptation Builds Complexity. Addison-Wesley, Massachusetts (1995)
3. Complex adaptive system, http://en.wikipedia.org/wiki/Complex_adaptive_system
4. Holland, J.H.: Studying complex adaptive systems. Journal of Systems of Science and complexity 19(1) (2006)

5. Zhang, S.: Study from Social Software, Web 2.0 to Complex Adaptive Information System (in Chinese). Ph.D Thesis, Renmin University of China (2006)
6. Wang, R., Fang, M., Chen, Y.: A Model of Collaborative Knowledge-Building Based on Web2.0. In: IFIP International Federation for Information Processing, vol. 255., Xu, L., Tjoa, A., Chaudhry, S.: Research and Practical Issues of Enterprise Information Systems II, vol. 2, pp. 1347–1351. Springer, Boston (2007)
7. Dynamic Systems Development Method, http://en.wikipedia.org/wiki/DSDM
8. Peter, M.S.: The Fifth Discipline: The art and practice of the learning organization. Doubleday, New York (1994)