14

Two-tiered Architecture of IS in WWW Environment

Yu Chen, Hui Han, Xiaofeng Wang
Information School, Renmin University Of China, Beijing, P.R.China
Tel: 086-10-62513256. Fax: 086-10-62513254.
Email: yuchen@public.bta.net.cn

Abstract

With the involvement of network, IS (Information System) becomes larger and more complex. No longer in the form of a central database with many application programs locating around it logically or physically, IS is aggregated by heterogeneous nodes. The two layers(system and node) are quite different in nature. And the nodes are active agents with their own resource and mechanism. Therefore, the system designer should reconsider IS using a hierarchical approach.

The CAS (Complex Adaptive System) theory is a nascent discipline provided by Santa Fe Institute at New Mexico, which emphasizes the role of Adaptation during the evolution of complex system. We have found that the CAS theory can provide a new perspective for IS development.

Based upon the CAS theory and the understanding of complex IS, we put forward a two-tiered architecture model for IS in WWW environment and

Information Systems in the WWW Environment C. Rolland, Y. Chen & M. Fang (Eds.)

argue some issues about the development methodology of IS. Our model is not complete yet and there are still many details to be discussed during the implementation. Here we only suggest a perspective to study the IS. We hope these new ideas will contribute to the IS development.

Keywords

CAS(Complex Adaptive System), IS, Architecture, Two-tiered, Agent, Life-cycle Methodology, Prototype Methodology

1 THE NEW FEATURE OF INFORMATION SYSTEM IN WWW ENVIRONMENT

With the involvement of network especially the emergence and popularity of WWW, we have stepped into a new society called information society. In such a society, extensive and massive information exchanges are going on nearly every minute among all kinds of entities in network environment, facilitated by the network infrastructure and other IT. [Here we refer to entities as all relevant commercial or noncommercial enterprises, corporations and other organizations.]

No longer self-closed and focusing mainly on internal business, any entity is just a node in the extensive-linked and dynamic network in both physical and logical sense. External relations and information thus become crucial to an entity's fortune even its survival. However, the external scene confronting the entity is so dynamic and unpredictable that the IS of the entity should adapt to the changing circumstances first and then be intelligent enough to handle so many semi-structured and unstructured problems.

Hence in response to such new circumstance and increasingly complex requirements from human being, the IS in the new era is more complex and more powerful in function and acquires a higher status and more extensive meaning comparing with traditional MIS.

We know that the traditional MIS serves in a specific organization and fulfills pre-defined tasks, managing mainly internal data with a unitary goal and an integrated architecture. Notwithstanding still serving a single organization, a contemporary IS is more active as an integrated complex system which has its own aim and mechanism in response to the challenging network environment. And besides the concept of traditional MIS, IS includes various information management systems like Commerce IS, etc. Meanwhile, different ISs may combine dynamically to aggregate into a larger IS in which case they join together when needed and quit from the group when mutual interests disappear.

As for a single IS, present-day MIS, for example, its subsystems or modules should no longer be the submissive parts of the whole IS and should acquire more freedom in respective business dealing.

It's not difficult to see that in whatever aspect of IS such as underlying infrastructure, scale, function, influence or IS development methodology, there exist diversity and complexity. But we can still capture the hierarchical feature of IS.

Based upon careful study, we consider the large IS in the new WWW environment as a complex adaptive system and regard Internet as a typical example.

Internet or information superhighway, or I-Way, or 'INet, as it is variously called, is in practice a global networked information system in the form of the WWW (World Wide Web) at the present stage of its development. We will then gain better insight into the Internet as a complex adaptive system. We could call Internet the largest IS. The following three distinct features of the Internet which may become the common features of the near future IS can lend credence to this viewpoint.

(1) Internet is a Huge-scale, Complex IS

It is composed of numerous large-scale or small-scale ISs connected with each other by networks throughout the world. These ISs have little in common whether in infrastructure, information resource, mechanism, function or others. However, the Internet as a whole works harmoniously with each individual IS functioning properly under different protocols; and its overall phenomenon is not irregular. Thereby, we can always draw a conclusion that the Internet is a complex system.

(2) Internet is a Self-organized IS

Internet is a superhighway on which all sorts of information are communicated without much intervention. In the past or at present, no one can imagine what appearance the Internet will take on and point out which direction it will go to. It can be seen from the short history of development of Internet that it was born and grew just like the organisms in biology. It is the result of the self- organization and self-development, instead of the purposed planning by any specific countries or organizations.

(3) Internet is a Hierarchical IS

There are significant distinctions between the entire Internet system and its constituent, local systems based on heterogeneous networks. As far as the local systems are concerned, they are operating independently, and have their own goals and features. When these systems are connected, however, they integrate into a whole Internet. This new tier takes on an entirely different look. It has features that are not the simple collection of individual networks.

The Internet has changed the world, including the way we look at the IS. More and more ISs based on networks appear today. How to understand their properties under WWW conditions and how to develop an IS in WWW environment; these new and critical issues need attentive study. When facing conditions different from those of the old days, what we need is a new theory and methodology that can address the issues the old ones can not explain.

2 THE THEORY OF CAS AND ITS ENLIGHTMENT ON INFORMATION SYSTEM DEVELOPMENT

CAS is the acronym of Complex Adaptive System. The theory of CAS is a new one on the law of complex system's development and cognition methodology. Its essential idea is that considering adaptation as the key factor, we analyze the mass information frequently interflowing between active agent and its environment so that we can recognize, simulate, manipulate and utilize the internal law of complex system development.

The origin of CAS theory can be traced back in the 1960s and 1970s when John Holland first put forward the GA(Genetic Algorithm) (see [1],[5]). Later in 1994, his report, "Hidden Order"(see [2]) made at Santa Fe Institute suggested the formation of CAS theory. Other scholars at SFI such as S.Kauffnian, J. Casti studied CAS theory from different angles and proposed their own arguments in this field(see [4],[5]).

The CAS theory turns out to be very beneficial to the development method of IS. Such enlightenment appears to be especially important to and suitable for the complex IS in the WWW environment.

The key points of CAS theory and some revelation are as follows:

(1) CAS theory defines Complex Adaptation System as the aggregate of many active agents. That is, CAS has two layers in architecture -----the system and the agent. Each agent can actively adapt to the environment by adjusting its behavior and altering its internal structure so as to attain its specific goal ultimately. We find that IS in the WWW environment possesses the similar two-tiered architecture. However, in the traditional MIS development we are usually confronted with an enterprise or an organization that has a unitary target and central control. Therefore, the architecture of IS is always constructed in a consistent way. Now in the WWW environment, the situation confronting IS is quite different. Every node in the WWW environment is independent since it has local interest and target and pursues local perfection. Every agent has its own architecture and mechanism while the whole system is a federation formed upon the interaction between the agents and their environments. Holland called such federation the ECHO model. Obviously, such idea of two-tiered architecture inspired us in the development of complex IS.

(2) To survive and develop in the special situation, every agent should possess such functions as receiving information, filtering information, responding to stimulus, accumulating experience and self-adjusting according to the experience. The theory of CAS brings about the new concept of chromosome and fitness, which lay the foundation of the logical model of active Agent. Holland defines chromosome as the material carrier of Agent's behavior(therefore the carrier of the experience). The effect of every chromosome's reaction to the external stimulus in turn affects the fitness of the chromosome whose value increases by success and decreases by failure. This

procedure of fitness change is called Credit Assignment. These points above construct a basic frame of the logical function for every agent. And these points are undoubtedly the crux in IS development.

(3) We are interested in the point of view suggested by the theory, i.e., the patterns of behavior reflected by the chromosome may be different, even contradictory. This point is just opposite to the strict demand for consistency in the previous research of IS. In the traditional IS methodology, we have tried to avoid or eliminate inconsistency in IS. However, the theory of CAS regards such inconsistency as an inevitable fact and melds it into the model. At the same time, such inconsistency is viewed as the foundation of agent's evolution and adaptation. The ECHO model demonstrates a concrete adaptation procedure by utilizing some of chromosome mechanism such as crossoner, reproduction and mutation. It's evident that the theory of CAS benefits much from biology and economics. Such problems mentioned above are emerging from the transition of traditional MIS environment to the WWW. They definitely need to be solved.

(4) The theory of CAS summarizes six categories of basic relationships among agents such as discernment, selection, resource transfer, adhesion, replication and reproduction. Obviously, there are two basic conditions to ensure these relations: the channel(or the way) and the standard(or the protocol) of information transmission. Therefore, we got the enlightenment that in macroscopic sense the large complex IS in the WWW environment is quite different from the traditional MIS. Its organization and implementation do not simply mean defining function, designing structure or organizing to realize, but creating environment, formulating the special dealing rules to guarantee the macro state of system development. The growth of Internet and the WWW is evidently a typical example.

(5) Viewing the whole procedure of IS, we can see that the IS has been growing continuously. Thus, the traditional IS development methods including the Life-cycle methodology and Prototype methodology are no longer convenient. We should reconsider some issues such as the growth of macro IS, its evaluation criteria and the development method of macro IS. We approve of the idea of Prototype methodology while there are still much improvement to do.

3 TWO-TIERED ARCHITECTURE ---- THE CONCEPTION MODEL OF LARGE INFORMATION SYSTEM IN THE WWW ENVIRONMENT

As computers distributed in different regions are connected into an integrated network, we obtain a platform for information sharing and processing. How to organize various information applications depends rightly on the conception model of such networked IS.

As we know, in the whole economic life at present there are various independent economic entities, each of which can be subdivided into divisions, each of which again is made up of subdivisions. Generally, these entities, divisions, or subdivisions are independent with their own benefits. Therefore, the

corresponding IS of an entity should consider well the hierarchy of the reality and guarantee that the corresponding child(i.e. sub) Information Systems which are in agreement with the distinct-interest-based divisions work harmoniously. Similarly, the larger IS such as the Commerce IS should also pay much attention to this point.

Then what kind of model should we build in the real IS construction? Inspired by the theory of CAS and observing well the reality, we recommend the hierarchical approach and put forward a two-tiered model as its embodiment. The model is not complete yet, but we expect the idea contained in it will do some contribution in the real IS construction in the new WWW environment.

The IS in our model has a two-tiered architecture with the system as the upper tier and the agent as the lower tier(see Figure1). While every agent works independently and interacts with other agents and the external environment dynamically, there is no rigid bondage among them. It's the laws or regulations that join them together into the whole system and regulate their activities through the stimuli. Therefore, every agent works with its own mechanism and adjusts itself in response to the external stimuli to adapt to the changing situations. And the whole system is coupled with these agents and acquires some new nature, maybe taking on an entirely new look in many aspects such as function, type and quantity of information, communication pattern, etc.. Figure 1 illustrates the two-tiered architecture of IS and the agent in it refers to the IS of an entity or its department or the federation of various entities.

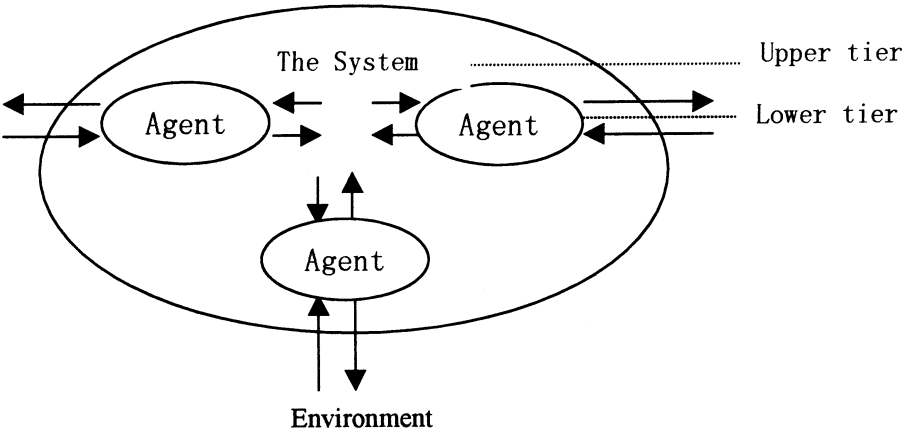


Figure1 Two-tiered architecture.

First of all, we will describe the kinds of IS, to which such a Conception Model is applicable:

(1) The agent is independent. Every agent has its own resources and independent mechanism and undertakes some special function.
(2) The environment of the agent is dynamic and filled with diverse and sometimes even messy information. The agent always encounters unpredictable problems

while the solution to the problem, the accomplishment of the agent's own goal and the implementation of the strategic decision depend much upon the environment and other likewise agents' activities.

(3) The relationships between agents are flexible. The agents influence each other, but the agents are not like the simple procedures in an assembly line, restricted in a sequential order.

- It is uncertain that which two agents establish relation with each other and when this relationship begins and ends.
- In the relationships between agents exist not only cooperation but also conflict of interest.

(4) The control between the system and the agents is indirect. The scale and function of the system is dynamic because the context and activities are continuously changing. And even the goal of system can't be described quantitatively either. So it's difficult to make an overall planning, identification of functions and design structure in the traditional sense.

(5) The concepts of system and agents themselves are also relative. That is, the system can be one agent in a larger system and the agent can be a two-tiered small system. Here we only suggest a new point of view.

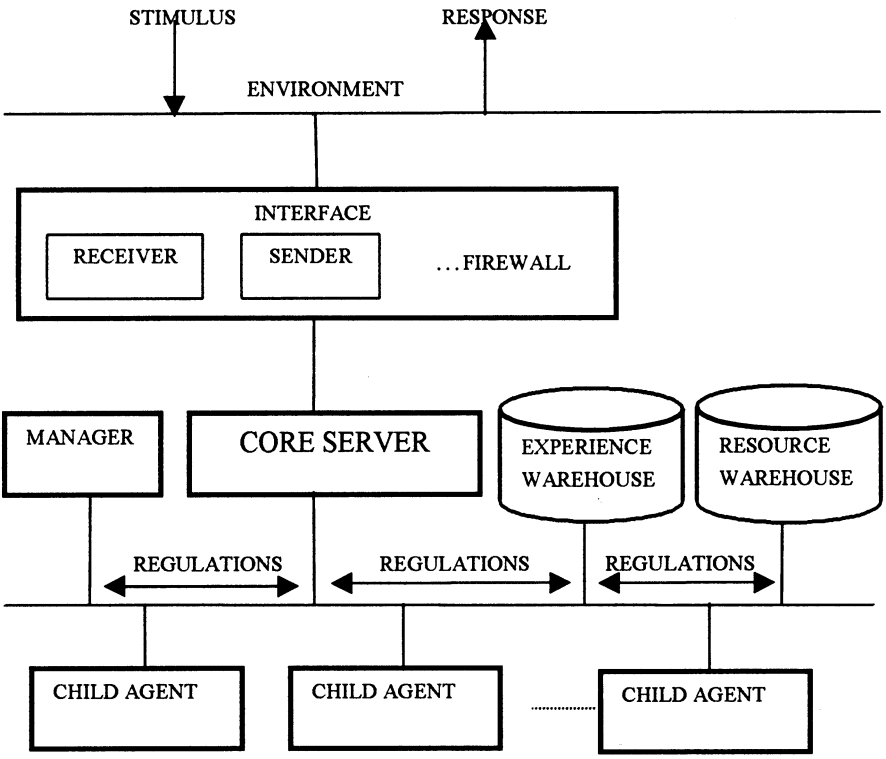
Someone may feel puzzled. Didn't the traditional MIS have the same two-tiered architecture of the whole and the part? Yes, they did. But after studying it carefully, we can understand that the traditional MIS has a clear specific target and task. Each part only undertakes one part of the whole task. Moreover, the logical relationship and information flow between agents are fixed in the traditional MIS. As to the agents, they have a low degree of initiative, with its own aim, which is part of, or directly related to, the overall aims of the system. They only collaborate to attain the whole target. When the IS has been established, its function and scale stabilizes, difficult to enhance or expand. So we can say that the traditional MIS has a static and fixed architecture.

Obviously, the traditional Conception Model is no longer suitable to the IS in the WWW environment, because they can't reveal the macro feature of IS. We can see that the traditional CM (Conception Model) whether Flowchart, Structure Chart, DFD, E-R chart, Petri Net or mathematical and logical model are only flat descriptions for the particular problem. For example, DFD serves for the description of the information flow in a specific business processing; E-R chart for the property of an entity and the relationships between entities; Petri Net for real-time control; the mathematical and logical model for the DSS. Each of such Conception Models has played an important role in MIS analyses and it is and will still be valuable in the future. However, IS in network situation is an aggregate of heterogeneous subsystems. The whole system grows dynamically with the birth and growth of the subsystems. Thus, the traditional CM can't describe adequately the growing procedure, the difference and relationships between the whole system and every agent. All these problems impel us to reconsider the IS from a new

macroscopic perspective. Therefore, We put forward the two-tiered CM as the embodiment of the hierarchical approach.

3.1 The agent

Here we define the agent as the autonomous and active entity. In reality, it may be the IS in an enterprise, a department of the corporation, or a special processing module, etc. It stayed in a dynamic environment full of messy information. The changing context brings so many unpredictable problems that the agent can't build a model in advance to solve them. What it can do is only accepting the stimulus from outside, and making rules immediately at that time and responding. Here we use the Internal Model to illustrate the mechanism of independent actions(see figure2).



NOTE: The CHILD AGENT here may refer to the CHILD IS with the similar structure in lower hierarchy, or the APPLICATION SERVER or END USER.

Figure2 Internal model of an agent.

3.1.1 INTERFACE

Nowadays, the pervasive networks and the WWW bring the IS a mobile circumstance. Each agent is interconnected with its environment with mass interchanges among agents and between the agent and its environment. Every agent should collect and process the diverse external information effectively to survive and develop. So we should pay much attention to setting up this interface. A good interface can help collect the information effectively and enhance the efficiency of the whole system's transaction.

Interface includes mainly two parts in structure: Information Receiver and Information Sender. Besides, there should be other parts like firewall to guarantee security, information filter, etc. It's clear that the main responsibilities the Receiver and the Sender should assume are to transfer information. But they should also perform some intelligent functions so as to lighten the burden on the CORE SERVER. For example, in the agent IS which serves a business firm, the Information Receiver should undertake such responsibilities as information filtering and classifying to obtain from the torrent stimuli it is inundated with what it needs such as orders, cash flow, incoming goods, and so on. As for the information sender, it may be able to respond to some simple stimuli itself, PUSH information regularly to the customers at their requests and seek information from outside or contact with other agents initially such as searching for a proper partner in agreement with the agent's intention.

3.1.2 CORE SERVER

This is the kernel of the agent. It receives filtered stimuli from the interface and analyzes it, and then makes responding rules and maybe breaks down the whole task into smaller jobs and assigns them to relevant child systems or dealing modules. When no outside stimulus arises, the CORE SERVER carries on its regular internal information processing just like what it does in the traditional MIS. To accomplish the agent's adaptation to the changing situations, the CORE SERVER can also adjust its behaviors and inform its child systems such variations so as to adjust and manage the child system's activities accordingly. If we view the agent as a human, then the CORE SERVER can be compared to the brain of the human.

The CORE SERVER includes mainly three kinds of actions as follows:

(1) Analyze problems
(2) Make the responding rules
(3) Enhance its adaptability

- Study the effect of the rule implementation, so as to adjust the rule's weight according to the feedback.
- Learn from the human and other sources.

(1) Analyze problems

The problems the agents encounter include not only external stimulus but also internal request. In order to deal with them effectively and give out proper response, the CORE SERVER should further analyze these problems. There are no ready-made solutions especially to the new problems. In this case, Problem analyzing appears to be very important.

Actually, there is no absolute boundary between Problem analyzing and Rule making. They complement each other since the result of Problem analyzing is the premise of Rule making and Rule making affects the pattern of Problem analyzing.

(2) Make the responding rules

Based upon the result of the Problem analyzing, the agent should search the experience warehouse for the proper problem-solving rule. If it exists, just select it and act according to it; If it doesn't, the agent should analyze the problems and the existing rules carefully, or try to figure out some new rule(s). Such new rule(s) will be added to the experience warehouse. There exist many candidate methods to make rules for a new problem. We can select the proper one in IS implementing. In CAS, the method of building block is introduced. Because every agent is faced with a perpetually novel world, it's unreasonable to prepare responding rules for every situation it may plausibly encounter. Thus we recommend Holland's method of building block. Because it's hard to explain it in just several sentences, we only introduce its main idea here. It suggests that we decompose the newly encountered complex problem into several familiar parts which then activate simultaneously the appropriate responding rules and we then combine these rules to figure out a new problem-solving rule(s). The whole procedure is just like breaking down the new problem into blocks and then building them in a new way. It embodies the agent's autonomy here. As to how to break down the problem and how to figure out the new rule, we can refer to the Genetic Algorithm introduced by Holland in his "Hidden Order".

(3) Enhance its adaptability

Its main job is to manage and adjust the responding rules in the experience warehouse. It is indispensable in the whole agent's working procedure because it determines the degree of agent's adaptiveness to the environment. There are two methods to accomplish it.

- Study the efficiency of the rule's implementation so as to adjust the rule's weight according to the feedback.

Because of the continuously changing situations, the rules in the warehouse may no longer be suitable and the new rules just made may also be improper. Therefore we should trace the utility of the rule, and then elevate the status of the effective rules and lower the status of the ineffective rules and sometimes even eliminate the outmoded ones. We assign each rule a strength or credit that, over time, comes to reflect the rule's usefulness in Problem solving. We Modify the strength of rules

according to their performance, in which case the whole procedure is called Credit Assignment in "Hidden Order". Just like the procedure of the Survival of The Fittest, We only keep the most promising rules in the Experience Warehouse according to their strengths so as to help the agent gradually adapt to the environment. There is a detailed account of the rule adjusting in Holland' Credit Assignment. We can use it for reference in IS development. Besides, we can also borrow some ideas form other theories or schemes.

- Learn from the human and other sources.

Actually, although the agents of the IS become more and more intelligent, they still can not compare with the real creatures in ecology. Therefore, the human capability of knowledge creation is much to be expected. We can design a Human-Computer interface here to accept the human's lesson. It can also improve its ability in other patterns.

3.1.3 EXPERIENCE WAREHOUSE

It should contain enough responding rules to the diverse stimuli. These rules can be called experience, which are accumulated in the long time contact with environment and other agents. These rules are very crucial to the agent's survival and fitness. To ensure the rules to be the proper solutions of the problem, there are always selection and adjustment according to their real effect. Every agent's warehouse can be special since each agent may serve a distinct organization, assume a specific task and be in a different condition. Taking function and efficiency into consideration, we should also design unique suitable responding rules and adjust them as occasion demands.

3.1.4 RESOURCE WAREHOUSE

It stores and manages the data, program and other resources.

We should make a good scheme to arrange the storage of the above two warehouses. We recommend the idea of Componentware or Component Software, which facilitates easy programming, reuse, and modification.

3.1.5 MANAGER and REGULATIONS

As we have explained in the previous parts, the agent even its child agent has its own working mechanism, thus exists among them only the loose relation. And such loose relation is maintained by the general regulations, which are about communications and some kinds of transaction collaborations whether commercial or noncommercial among child agents. So it's important to make proper regulations and adjust them according to the requirements of function and management variation or the requirements from the CORE SERVER. All of such

responsibilities belong to the MANAGER.

3.2 The system

IS in the network environment is the coupling of autonomous and active agents, but it isn't the simple gathering of the agents. Uniting the agents to achieve the whole system's goal, the system should create a good environment to guarantee that the agents work harmoniously and autonomously.

We emphasize the importance of making rational regulations to accomplish the macro control and good management.

3.2.1 There are two kinds of regulations such as the communication regulations and the transaction collaboration regulations among the agents

1. The communication regulations or protocols

It's the essential condition that we ensure the communication among agents. Till now, there have been many rules such as the protocol of TCP/IP which become the de facto standard of communication, HTTP which is now the basic protocol of Web, HTML the grammar of the WWW document... It's feasible that we deploy special communication standards in respective agent. However, it's important that every agent knows clearly the public protocol to correspond with each other.

In an ideal way, we can adopt uniform communication regulations among every agent. But considering the complex situation and the heterogeneity of the systems, it's not feasible to stipulate only one protocol and force all the systems to observe. We hold that it's good to employ a mechanism similar to the middleware. That is to say, we can use a layer that can tolerate the different systems and transform them into the receiver's pattern. This idea is inspired by the middleware.

The CORBA (Common Object Request Broker Architecture) scheme can be a good example. It bridges the chasm between heterogeneous components which are unable to talk to each other directly. The Client component can talk to ORB, an intermediary (broker), through the common language IDL (Interface Definition Language), and then its request for services, written in IDL, is mapped into the host language (i.e., C or C++) of the server.

Of course, TCP/IP is a network protocol, or in terms of common language, it is the traffic law of the (information) superhighway. A middleware such as ORB operating under TCP/CP acts as an intermediary for the APPLICATIONS (or components or objects) to communicate with each other. Network protocol and middleware each belongs to the different level of the network architecture. Here we just take them for examples to explain some ideas we can use for reference.

2. The transaction collaboration regulations

While the communication regulations just offer the channel for agents to contact with each other, the transaction collaboration regulations allow them to collaborate

to do some profit or nonprofit business such as trade, long-distance education through the network, or network game for multi-players. The EDIFACT in Electronic Commerce and the rules of network game and so on can be viewed as such regulations.

3.2.2 The function of regulations

We hold that it's very important to make appropriate regulations to guarantee the well operating of the whole Information System. As follows we list three of its advantages as accentuation.

Firstly, it helps achieve the autonomy and adaptation of the agent IS.
We can see that the individual agent is just like the economic entity and the regulations are like the laws or rules in the market economy. Every agent works independently and aims at its own maximum benefits, while in the mean time it abides by the regulations to contact with others and they collaborate to do some transactions. There are no inter-operations between the internal parts of any two agents who collaborate. As Figure3 illustrates, they collaborate based upon these general regulations, therefore ensuring the autonomy of the agents.

Also, there are few direct commands to the agent controlling or altering its behavior. It's the variations of the regulations that act as the external stimuli to the agent and the agent adjusts itself accordingly to adapt to the new situation.

Secondly, it facilitates the functional adjustment of the whole system.
Because regulations are generally the independent part of the whole system and independent from the agents, we can only change the regulations themselves if the functional adjustment of the whole system has a bearing on the regulations. And so long as the interface of the regulations is kept invariable, there is no great adjustment necessary in the whole IS. As explained in the first advantage, the agents also adjust their activities according to the variations of the regulations, so as to help accomplish the functional adjustment of the whole system.

Thirdly, it allows the growth of the whole system.
Just as people speak according to language grammar, the agents of the IS should first abide by an established standard to contact with each other and do some dealing with others. We find that the stable grammar can still accept the new words and phrases thus enriches the language. Likewise, the regulations should be consistent and stable, thus give other agents the opportunity to join into the system. Therefore, the system expands provided that the newly joint agents observe the communication and business protocol.

3.2.3 There exist various relationships among agents of the IS in the network environment just like the humans in a society. We group the various relationships into the job division and the cooperation between agents

Job Division: Reflecting the reality in our society, every agent undertakes specific function and stores certain amount of data. This makes the whole system to be in the form of function distribution and data distribution. Such status can be called Job Division.

Cooperation: The various complex relationships among agents are fully manifested in their cooperative pattern. Calling it cooperation just means that the agents are independent elements above all and at the same time with dynamic relationships among them. They start and end their cooperation freely. We build a model of Request/Reply to illustrate such cooperation schematically(see Figure3).

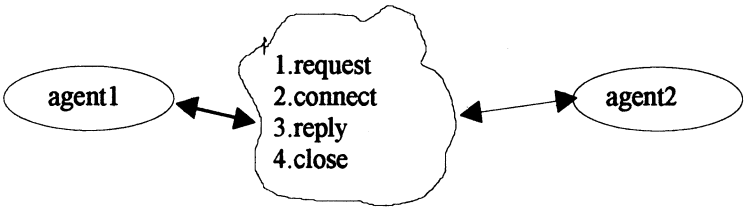


Figure3 Request/Reply model.

Agent1 should request Agent2 first and they consult each other before finally establishing the cooperative relationship(i.e. connect). Only when the relationship is built, can they start the interchange procedure. And they disconnect when the collaboration is over. This is the idea of TCP/IP. That is to say the relationship needs not to be kept forever once the relationship is built.

No matter how complex the logical relationships between the agent1 and agent2 are, they are encapsulated in the model as the flows between the agents, whether the request is polite asking or demanding, or the reply is some static data or some processing or in the form of a dynamic agent to interact with the client continuously .

However, one agent can't interfere with the other's internal processing. It can't manipulate other's data directly. In the above example, what agent1 can do is only sending requirement to the agent2 and getting the reply. The agent2 is really in charge of the real processing. Thus the independence of each agent and the realization of the coupling of agents are guaranteed, which also provide some opportunity for control distributing, parallel processing, etc. It also facilitates the entrance and withdrawal of agents and the adjustment of their relationships.

Coordination. There are conflicts among agents because they have their own interests and dealing pattern, which require the macro coordination.

We should consider thoroughly the formulation of protocol, job division and the agent's internal model, and utilize the existing fruit of research to help accomplish the proper coordination.

The above introduction of two-tiered Conception Model is not the result of the simple analysis of and an isolated, static view of the IS. It comes from the deeper understanding of the character of the IS in the network environment and its history of development..

4 THE TWO-TIERED ARCHITECTURE MODEL AS A GUIDANCE TO THE DEVELOPMENT OF INFORMATION SYSTEM

It can be seen from the discussion above about the two-tiered Architecture model--a new conceptual model--that, when we use this model to study the complex IS in the WWW environment, this model is not only compatible with the existing system well, but also provides a new perspective to develop the IS. We can understand the IS profoundly and precisely by the help of this conceptual model. Meanwhile, we can also get some guidance from it to the development of a new IS in the network environment. It may be premature to assert that the two-tiered Architecture model can bring about innovative changes in the development methodology of the IS, but the novel ideas contained in it will help us even when we use existing methods to develop the IS in the network environment.

Now that the new conceptual model provides us with a hierarchical method to analyze the complex IS, when we develop a new IS, thinking over the characteristic of these two tiers and their relations carefully, we will get substantial enlightenment.

4.1 The Continuous Process of System Evolution

The Two-tiered Architecture model emphasizes the stimulus-response mechanism of the information system which is constantly adaptive to the environment. In fact, this emphasis puts the weight on the dynamic property of the system development. Because a system is always in a continuous evolution process and grows in this process, understandings of a system should follow its developing process, but not focus on its certain static intersection.

As we realize that the whole system of the complex IS evolves continuously, somewhat like the natural system in the biology, we hold that it is not good, especially at the early stage of development, to develop IS according to the fixed design in which the final state of the system is stipulated in advance and allows no changes. The more complex an IS is, the less feasible this approach can be. The methodology such as Life-cycle one in the conventional sense is deficient at this point, since it needs much more rigid design procedure and refuses to change frequently. However, the dynamic thought of the Prototype Methodology is more applicable to the development of complex IS.

4.2 Adaptation--the Key Factor to System's Evolution

The Two-tiered Architecture model emphasizes the importance of the environment where a system exists. It accentuates that the environment is the driving force of

the development, growth and evolution of a system. A system receives stimuli from the environment, reacts in response to them by adjusting itself automatically, and evolves accordingly.

Now that the IS under discussion is a complex system in the WWW environment and lives in a dynamic, changeable environment, its ability to adapt to the environment is even more indispensable.

This constraint the environment imposes on the IS must be considered carefully when we develop a real IS in the environment. In practice, we must assure that developed IS is not a rigid, static, "dead" system, nor is it a self-closed system. We should ensure its openness, scalability, dynamic adaptation and intelligent reaction to the environment.

In fact, this is an issue involving system modification and maintenance in the later development stages. Some specialists have estimated that approximately seventy percent of the expenditures and resources are spent on modifying system function and maintaining system's normal work. This is partially due to the inadequate contemplation and estimation of the effects that environment impacts on the system. The result is that developers had to modify its function entirely by manual work in order to achieve its goal. As the system has become more and more complex, it is not a scientific way in doing so since it will cause great waste of the precious and limited resources.

An IS is a livable one only when it is an open and adaptive one. We should keep this conception of openness in mind at the early stage of development, especially at the logic design stage. We should take into consideration the environment's effects and system's openness as thoroughly as possible.

4.3 The Autonomy of the Subsystem

In the two-tiered Architecture model, Autonomy is an important characteristic of the subsystem. A subsystem is an autonomous, active entity, though it has many relationships with the whole system and other subsystems.

Each individual subsystem has its distinguishing characteristics depending on the environment it resides in, the relationships it has, and the reactions it responds to the environment, etc. It is necessary to think over these different characteristics when an IS is to be developed, and adopt a specific development method appropriate to a specific subsystem accordingly.

That different subsystems use different methods is justified by the agent's characteristic of independence. It is just because of the existence of this independence that different subsystems using different development methods will not affect and will not be affected by the development of the whole system and other systems. In addition, the relationships between subsystems, as well as those between the whole system and subsystem, will not be affected, either.

Thus, in the actual work of development it is not necessary to stick to one particular method in all the cases. One method that is applicable to some subsystem may be not apposite to another. While in the case of the system which is in dynamical growth, the Prototype Methodology may be used in an incremental way

of refinement, the Life-cycle Methodology may be used cost-effectively and efficiently to develop a system which is relatively stable and exists in a simple environment. In a word, use the method which is most efficient for the individual subsystem, without enforcing conformability with the method of developing the entire system. This is the new idea revealed to us by the Two-Tired Architecture. Of course, we should not neglect the cooperation and coordination of the whole system and individual subsystems when we emphasize the importance of the independence of the subsystems.

4.4 The New nature of the whole system

The two-tiered Architecture model points out the distinction between the whole system and subsystems when discussing the hierarchy of the system. Different tier has different characteristics.

When we try to understand an IS, especially a complex one, the conventional approach we take is to decompose it into several layers, analyze the most easily comprehended, most simple layer with regard to the specific request of the question domain the system exists in, and then study the upper or lower layers along with the acquired knowledge of the analyzed layer. The lower layers are usually the simpler layers, and the study often starts at these layers.

When we understand the lower layers thoroughly, whether we can be sure to grasp the essence of their upper layers or not, is still a question. The upper layer, which is the aggregate of the lower layers, has the new characteristics that can not be understood by summarizing the knowledge of the lower layers simply. As far as the complex IS is concerned, the features and functions of the whole system can not be inferred or concluded only from the features of the subsystems. The new qualities need our special cognition.

Therefore, we must grasp the essential quality of the whole system while we scrutinize each subsystem. In particular, we should try hard to understand what new qualities will be produced on the whole system tier in the complex IS by the integration of different, independent and active subsystems. It's a difficult task, but it is worth our painstaking study.

5 CONCLUSION AND OPENING ISSUES

As the expanding worldwide network provides a new information platform for IS, we have to reconsider the working mode of IS and its development methodology.

Inspired by the CAS theory and studying the reality carefully, we regard IS in the new WWW environment as the complex adaptive system whose function, structure, and development method etc. are very different from that of the old IS (esp. MIS in the traditional sense).

Although the CAS theory is a newly developed theory whose concept and methodology are still under development, it brings us many novel ideas and grasps the common feature of many urgent issues in the modern society--complexity and adaptation. Not surprisingly at all, it draws extensive attentions from the academy

and application fields. And the CAS theory indeed gives us much enlightenment in recognizing and designing of IS, and its development methodology as well.

As follows are five aspects of our ideas and some opening issues:

(1) The large IS based on the WWW environment has a complex two-tiered architecture. Its substratum is composed of many independently and actively developing agents (i.e. nodes). The whole IS is formed by the interactions and inter-affections among these agents.
(2) The functional structure of each agent can be summarized to six basic points as following while in real implementation they may be put into different positions:
Receiving information
Filtering information
Choosing the responding rule(s)
Responding
Evaluating the result
Adapting and changing.
(3) Each agent can develop according to the fundamental principle of Prototype Methodology; its real mechanism can be constructed by referring to the Genetic Algorithm. However, there are many differences between biology and IS. Unlike the mechanism of agent in biology, which relies mainly upon self-effort, a real IS needs much more human interference.
(4) As far as the whole system is concerned, the traditional Life-cycle methodology and Prototype methodology are no longer convenient because it is difficult to make sure of the action of each agent in advance and make a fixed global schedule at the beginning. The critical factor for such kind of system is to provide the communicating channel or mechanism, and define the rules and standards of information exchange and transaction collaborations.
(5) In real IS implementation, much attention should be paid to distributed computing since IS is in such a network environment. Although extant Client/Server is proved to be an efficient mode for distributed computing, it serves well only the internal information sharing and application of an organization with unanimous or at least relevant interests. How to construct a rational hierarchical computing model achieving the information application among distinct-interest-based entities is worth much of our efforts.

That IS extends into the WWW environment raises a series of changes in IS architecture, its function and methods of its development and organization. These changes just start to unfold; there are not integrated and systematic solutions yet. Thus it's very necessary that the scholars and engineers work cooperatively.

6 REFERENCES

M.Waldlop,"Complexity",Ch5. SDK Joint Publishing CO.1995.
Holland.,"Hidden Order-How Adaptation Builds Complexity".Addison-

Wesley, 1995.
S.Kanffman, "At home of Universe" 1996.
J.Cast, "Complexification-A Science of Surprise".
M.MITCHELL, "An Introduction to Genetic Algorithm" MIT Press 1996.
Zhongzhi Shi, Wangwen Jie, QijiaTian, "The Current Status and Trend of Intelligent Agent Development", China Computerworld, Issue.4, 1998.
Shuo Bai, "The Strategy of Intelligent Agent", China Computerworld, Issue.4, 1998.
Yongshao Ruan, Wang Yu, "The Social and Ecology Model of Multi-Agent System", China Computerworld, Issue.4, 1998.
Zhaohui Wu, Yunhe Pan, "The Application of Multi-Agent System in The Management of Cooperation Quality", China Computerworld, Issue.4, 1998.
Aimin Wang, "The Techonology of Agent and Distributed Computing", China Computerworld, Issue.4, 1998.
Kris Jamsa Ph.D, Suleiman "SAM" Lalani, Steve Weakley, "The Course of Web Program Design", Publishing House of Electronics Industry,
Yu Chen, Shunyu Piao, "Management Of Information System", Publishing House of RenMin University, 1995.
Yu Chen, "The Introduction of Economical Information System", Publishing House of RenMin University, 1996.
Hongmu Tao, Qisan Pan, "Life Cycle of The Object and Its Application in The Analysys of Object in Information System" "The Theory and Practice of System Project", Vol.17, October, 1997.

7 BIOGRAPHY

Yu Chen is Dean of the Information School and a Professor of Computer Information Systems at Renmin University. Among the ten books he authored or co-authored are An Introduction to the Economics of Information Management and A Dialogue About Systems, both published by Renmin University. He authored over 50 articles that appeared in journals in China and the US. He has been an educator his entire career that includes consulting projects with major Chinese SOEs. His research areas are information economics, system science, and management information systems.

Hui Han and Xiaofeng Wang are currently postgraduates at Information School of Renmin University, Beijing, China. They are presently engaged in the study of the CAS theory and computer simulation on economics. Both of them are also the developers of SIM_EC. And they have participated in the development of information system of a real estate corporation and published two articles about e-commerce in a journal of china.