# RESEARCH AND DESIGN ON DOMAIN-AGRICULTURE-CROPS SOFTWARE ARCHITECTURE ORIENTED ADAPTIVE MODEL

Huanliang Xu[1],Haiyan Jiang[1,3],Shougang Ren[1,*], Xiaojun Liu[2,3], Weixin Cao[2,3]

*[1] College of Information Science and Technology, Nanjing Agricultural University, Nanjing 210095, P. R. China, Email: huanliangxu@njau.edu.cn*

*[2] College of Agriculture, Nanjing Agricultural University, Nanjing 210095, P. R. China*

*[3] Hi-Tech Key Laboratory of Information Agriculture, Jiangsu Province*

** Corresponding author, Address: College of Information Science and Technology, Nanjing Agricultural University, 1 Weigang, Nanjing Jiangsu Province, 210095, P. R. China, Tel:+86-25-84396350, Email:rensg@njau.edu.cn*

Abstract： Composing application with plug & play (P&P) agriculture-crop business component on the domain-agriculture-crops software architecture (DAcSA) is an ideal implementation mechanism to develop the domain-agriculture-crop applied system. A black and white box framework for the adaptive DAcSA is built based on the agriculture-crop business component and hotspot subsystem. According to the domain-agriculture-crops rules, an administer center was designed to realize the plug and play of business component in domain framework by gluing component and hotspot subsystems up, which deposit in component lib and hotspot repository respectively. In line with the domain-agriculture-crops information characteristics, a resource-model -analysis (RMA) data mode and corresponding behavioral model based on the adaptive DAcSA was proposed and a supporting system based on virtual machine architecture was built, which has been applied to the prototype development of wheat growth simulation and decision-making supporting system. The proposed adaptive model proves preferable adaptability and can effectively decrease the cost of development and maintenance.

Keywords： Domain-specific software architecture, Domain-agriculture-crops adaptive software architecture, Business component, virtual machine, Wheat growth simulation, decision-making supporting system

# 1. INTRODUCTION

Domain-specific software architecture (DSSA) is the core asset for domain-specific software development (Mei & Shen, 2006). It has the following features (Huang et al., 2006):  strictly-defined problem area and solution area;  domain-proper abstraction;  domain universality applied to specific application development in the domain; and  fixed, typical and reusable software element in the development process of the domain.

Research on domain-agriculture-crops software architecture (DAcSA) oriented adaptive model is to explore the software system construction and adaptability in the domain of agriculture crop. As the fundamental and principle research in agriculture information, it involves the digitalized expression, design, control and management of domain- agriculture-crops objects (such as biological factor, environment factor, technique factor and social economy factor) and the process (such as growth process, service process as well as management process etc.) (Yu&Cao, 2004). DAcSA mainly designs a universal and agile software architecture oriented to agricultural-crop domain, and eventually creates an application software system effectively and automatically (Cao et al., 2006). Therefore, this article will emphatically study a software process model to adapt for the domain- agriculture-crops and the corresponding architecture to support this process.

The research on numerous domain-specific software architectures (for example, self-adaptive intelligent system, aeronautic electronic equipment system etc.) have been carried out with according achievements abroad. The domestic DSSA research mainly concentrates in EIS (for example, tobacco, insurance, city geology etc.) as well as the intelligent systems (Barbara et al. 1995, Li & Wu. 2005, (Shang et al., 2006). At present, it has two kinds of realization methods about the DSSA. That is: Compile- time method, which creates the executable code according to the software system model automatically; and Run-time method, which runs the software system model in the Virtual Machine (VM) directly. The former has realized the automatic creation of code, but it is costly and disadvantageous to the supporting system development. Moreover, it is difficult to manage due to the numerous codes created automatically. The latter is implemented based on VM, thus it only involves the description of system script model with less code quantity, which is in favor of the code management and maintenance. In the article a DAcSA-oriented system model called “resource-model-analysis (RMA)” is proposed, taking the digital agriculture-crops as an example. Furthermore, an RMA-based DAcSA is designed to meet the need of changeable, flexible and self-adaptive intelligent agriculture information system by a combination with virtual machine structure.

# 2. DACSA-BASED ADAPTIVE SYSTEM DEVELOPMENT

## 2.1 Formal description of the DAcSA-based adaptive Model

DAcSA-oriented adaptive model must keep system logic function properly under different environmental conditions, and could covert the passivity of entity unit to initiativeness, the homogeneous coordination mode to diversity and static system evolution to dynamic one. Herein adaptability means software system can adjust its construction and algorithm automatically along with the change of its operating environment, and achieve a balance between them in an evolutionary way. To facilitate the description of DAcSA model, this article gives some descriptions as follows.

### 2.1.1 Domain-agriculture-crops Business Component

Domain-agriculture-crops business component is relatively fixed, typical and reusable software element in the domain-agriculture-crops. According to the definition by W.Koyacyuski (Kwozacznski, 1998), business component is the software business objects. Domain- agriculture-crops business component is the domain-agriculture-crops business object component. It is the autonomous domain-agriculture- crops business concept and reusable software unit, formed during domain-agriculture-crops software implementation. And it may be the domain-agriculture-crops object or the set of domain-agriculture-crops objects or the structure involved some domain-agriculture-crops objects, which can implement some specific functions. Hereinto, domain- agriculture-crops business entity component and data component are two basic components in domain-agriculture-crops.

**AcEntity component**

AcEntity component is the set of business logic functions, used to depict any significant objects such as the static entity, dynamic event and processing logic. AcEntity would be described as follow.

$$\text{AcEntity} ::= \{\text{Name}, \text{Func}, \text{Code}, \text{Interf} | (\text{In}, \text{Out})\} \quad (1)$$

Of which, Func means component functional description; Code is the component binary target code entity; Interf is the component interface, including input interface and output interface.

**Data Component**

Data Component would transform the AcEntity component attribute into the data and store them in the database. That is, Data Component will execute the data processing function including data definition and data manipulating such as data storage, inserting, deleting, updating and so on.

$$\mathrm{Data} ::= \mathrm{Co}m(\mathrm{AcEntity}, \mathrm{Val}[, \mathrm{Val}]...) \tag{2}$$

$$\mathrm{Com} ::= \{Define, Store, Manipulatio\mathrm{n}\} \tag{3}$$

### 2.1.2 Domain-agriculture-crops Software Architecture, DAcSA

DAcSA is the mechanism or the framework for configuration agriculture-crops business components, and the channel network for connection of the information flow and the control flow among components. Usually, the framework could be divided into white-box framework and black-box framework according to the framework extension and customization technique. To take both of their advantages, the black and white box mixed framework technology is used to construct DAcSA, that is, the fixed part would be designed in the form of component (black-box) and changeable part would be design in the form of hotspot subsystem (white box) based on the design pattern (Xu & Li. 2003, Li & Xu, 2005). It is illustrated in figure1.

In Figure 1, The DAcSA includes the management & glue center, black box business component library, white box hotspot knowledge library and business rules library. In the “black and white box” mixed framework, the fixed spots in DAcSA framework are designed to generic business component and saved in the business component library and the hotspots designed to hotspot subsystem based on the design pattern and saved in the hotspot knowledge library. The domain-agriculture-crops business rules are extracted and stored in the domain regular library. The management glue center would be responsible for selecting constructing unit from the business component library and the hotspot knowledge library respectively according to the business rules in the domain rules library, and plug & play upon the DAcSA to complete the realization and running of business logic.

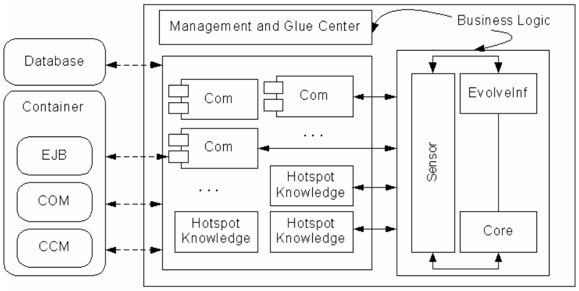


*Figure 1.* Adaptive model: black and white box mixed DAcSA

### 2.1.3 Agriculture-crops applied system, AcSytem

In accordance with the above specification, the self-adaptive AcSystem can be manifested by a unit triplet as follows:

$$\mathrm{AcSystem} ::= (\mathrm{DAcSA}, \mathrm{Businessrule}, \mathrm{Controller}) \qquad (4)$$

In formula (4), ① DAcSA framework is a set of agriculture-crop fundamental component and hotspot subsystem and management glue center; ② Businessrule, i.e. agriculture-crop business logic set, is the agriculture business rules set extracted from the agriculture domain analysis; and ③Controller means to apperceive and adapt business logic change and complete the evolution or instantiation of business component and the hotspot knowledge in DAcSA. Controller may be shown in the following unit triplet,

$$Controller ::= \{Core, \mathrm{SensorInf}, EvolvedInf\} \qquad (5)$$

Of which, Core means the logic structure of component for evolution; SensorInf means outside change information or input information. SensorInf deduces conditions triggering component evolution; EvolvedInf means the necessary evolution condition of the component. EvoledInf will justify whether the SensorInf triggers component evolution or designed for hotspot subsystem.

Conceptually adaptive DAcSA has three abstract layers, namely requirement layer, service layer and operation layer, to map the environment change and dynamic property to the software realization. Specifically, user build a requirement model firstly and configure management mechanism as shown in Figure 1 to realize the dynamic mapping of requirement and service.

## 2.2 Adaptive DAcSA-based RMA Model, DAcSA-RMA

### 2.2.1 RMA Model

Adaptive DAcSA design would support the AcSystem production and system evolvement with requirement change. In AcSystem, the system main body is agriculture-crops simulation model. Crops simulation model is the model simulated by means of the crops key composing element during its growth, which the crops get natural resource and grow according to the ecophysiological and zoological rule, as shown in Figure 2. It illustrates the general concept framework of the crops system simulation, key sub-model and simulation element and the interrelation- ship between them. Hereinto, A-F presents the 6 sub-models respectively. A is the development phase sub-model; B the biomass production sub-model; C the partition sub-model; D the organ formation sub-model; E the soil-water balance sub-model; F the soil-nutrition balance sub-model. And they are all restricted and influenced by climate state, conditions of soil, breed, cultivation and management.

Crop simulation model is composed of some independent sub-model, status or velocity variables and process function, which will get a simulation of a given subject (Wang et al., 2002). The ordered set of these concepts forms the whole process of crop simulation. The different crops model construction is the collection of the different algorithmic under the same or similar concept model. The running foundation of these algorithms is the set of various data.

Of course , although the simulation means, methods and the decomposition degree may vary, their goal or destination of simulation is consistent. Taking crop development phase for instance, the aims are all to simulate the growth process and predict phenology. There are different methods and models for different crops and different model- construction staff, but the main concept process and data model are similar (Yan et al., 2000). For example, in the wheat growth duration simulation, the temperature effect, light effect, jarovization effect and the interaction among every influence component are all considered to simulate wheat growth process and to predict phenology. For rice, because of no jarovization effect, only the temperature effect, light effect, growth process and predict phenology are considered to simulate the rice growth process and data. Meanwhile, both of them will also take into consideration of the restriction and influence of climate, breed parameters and cultivation condition.

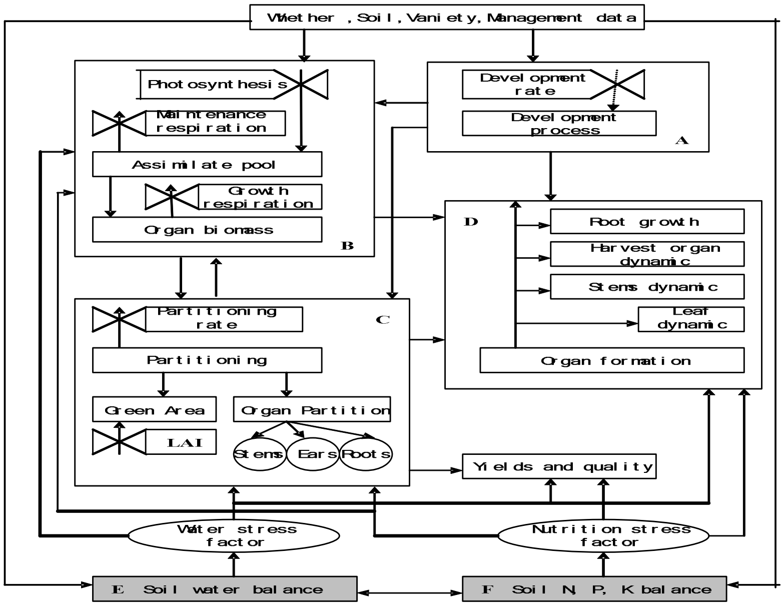


*Figure 2.* Generic Concepts and Framework for Crop Simulation

From the analysis given above, the key to build DAcSA is to extract the relatively fixed business component and the relatively volatile hotspot subsystem from different models for different crops and different model-construction staff. It is operation to data, whatever the abstract mode. From such viewpoint, DAcSA-based AcSystem is a status machine in nature. Therefore, it can also be marked as ：

$$AcSystem = \{I, O, S, \delta, \lambda\}$$

Of which , I is input set; O output set; S the inner status set in the AcSystem; δ status mapping function; λ output mapping function, and marked as:

$$\delta : I \times S \to S$$

$$\lambda : I \times S \to O$$

To analyze the data characteristics of crops simulation model further, there exists two kinds of data type sets with common character in the crops simulation model, one is describing resource data class (R), and the other is describing resource activity mark data class. The former is an objective resource or experiment-proved public data class, the latter is the data class result from analysis, reasoning and illation activity. In consideration of the characteristics of domain-agriculture-crops, this paper would separate the describing resource activity mark data into two types, simulation model data (M) and analysis data (A). It is called RMA model. In details:

R: resource data, it is the mapping result of outside natural environment entity on AcSystem, including environment data, cultivation and management data and factual measured data of the crops growth, e.g., the data about climate, soil, breed, cultivation condition and water and fertilizer management etc in different area;

M: simulation model data, it is the mapping result of the AcEntity and interrelationship between them on AcSystem, e.g. independent sub-model, status or velocity variables and process function and breed parameters related to simulation model, etc；

A: analysis and evaluation data , it is the mapping result of outside environment, human thinking activity or the change of user demand on AcSystem, e.g. the model prediction result data, the evaluation methods on crops etc.

Obviously, mapping relationship exists in R, M and A data in AcSystem, e.g. there must be R in M and there must be M and R in A. The data relationship among R, M and A can be marked as：

$$\Gamma(R, M, A) = \prod_{M \to R}, \prod_{A \to R}, \prod_{A \to M} \qquad (6)$$

$\Gamma$ means the mapping relationship among data.

$\prod_{M \to R}, \prod_{A \to R}, \prod_{A \to M}$ represent relationship of data M to R, A to R, A to M respectively.

On the basis of the RMA data model, its behavioral model can be further analyzed. Only from the viewpoint of structure, AcSystem behavioral model can be viewed as the set of three parts, i.e.

$$AcSystem = \Lambda\{M_R, M_M, M_A\} \qquad (7)$$

Hereinto, $M_R : R \cup M \to R$ means R class data processing part, which would be processed for R class data itself, or M class data would be processed into the R class data;

$M_M : M \to M$ means M class data processing part, which would be processed for M class data itself;

$M_A : (M \times R) \cup A \to A$ means M class and R class data would be processed into the A class data, or A class data would be processed into A class data itself. As shown in Figure 3.

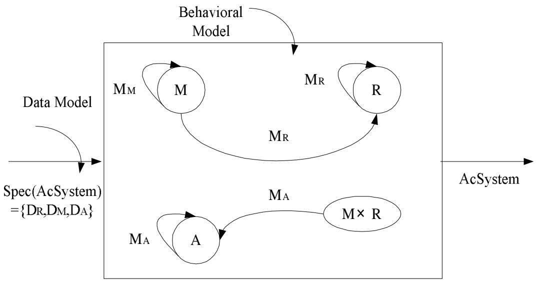


*Figure 3.* Behavioral model

Now take a wheat growth simulation and management decision- making support system as an example to explain RMA model. It can be described as:

R = {{soil parameter}, {climate data}, {variety description}, {cultivation condition}, {water-fertilizer management data}, {observed values in field }}…

M = {{variety parameter}, {wheat development phase sub-model}, {wheat biomass production sub-model}, {wheat dry matter accumulation and yield formation}, {wheat organ formation sub-model}, {wheat water balance simulation}, {wheat nutrient (N, P, K) balance}, (weather environment simulation)… };

A = {{model prediction data}, {index of strategy evaluation}, {project evaluation result}, {observed sensitive values}, {variety parameter debug}, {real-time prediction}, {temporal spatial analysis}, …}

Well then, the data mapping set among them,

$\Gamma$ = {{ wheat development phase simulation} → {weather, growth condition, variety parameter, water balance simulation, nutrient balance simulation}, ……{project evaluation result} → {M}, {sensitive observed values} → {model prediction data, observed values in field}, {variety parameter debug} → {R, variety parameter}, {real-time prediction} → {R, M, model prediction data, observed data in field}, { temporal spatial analysis } → {weather, soil, variety description, growth condition},etc.

Among the above, the behavior model $\Lambda$ for the project evaluation result can be described as follow.

$$\Lambda = \{M_A | R, M \times R, M_M | R, M\} \tag{8}$$

In formula (8), the behavior model for the project evaluation result includes the processing of relevant resource data and model data. These behavior models embed by domain-agriculture-crops business rules is mapping to software implementation. Management glue center (in Figure 1) would select AcEntity business component or hotspot subsystem to be instantiated from hotspot knowledge and produce AcSystem.

### 2.2.2 **Virtual machine framework-based system Implementation**

This article uses virtual machine framework to realize RMA. The virtual machine framework is constructed by the hierarchy pattern, and the data mapping layer, business logic layer, interactive control layer, interface presentation layer, the regulation control layer and the inner structure of regulation mapping layer would be design by the interpretation machine pattern (Guo et al., 2004). The inner structure of the virtual machine is as illustrate in Figure 4.

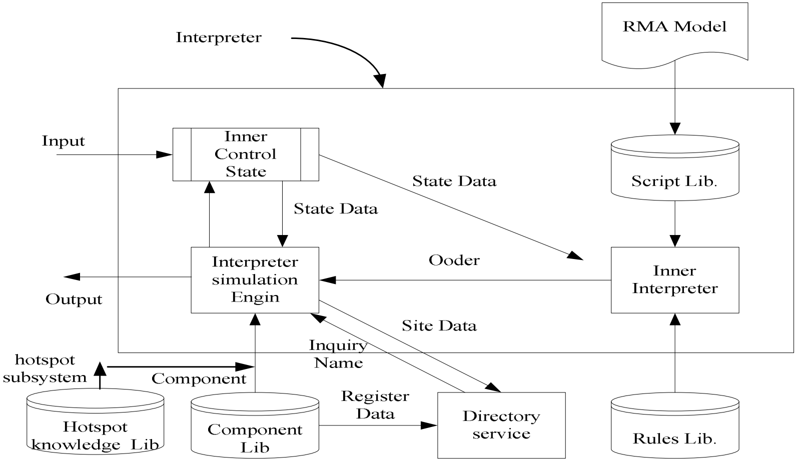


*Figure 4.* Inner structure of the virtual machine

Implementation process is as follow:

(1) Agricultural experts and software engineers build RMA model together;

(2) RMA model would be transferred into script form. Inner interpreter gets and checks the validity and correctness of the script according to business rules from rules library. Then instruction will be sent to inform the interpreter simulation engine in line with the script semantic content.

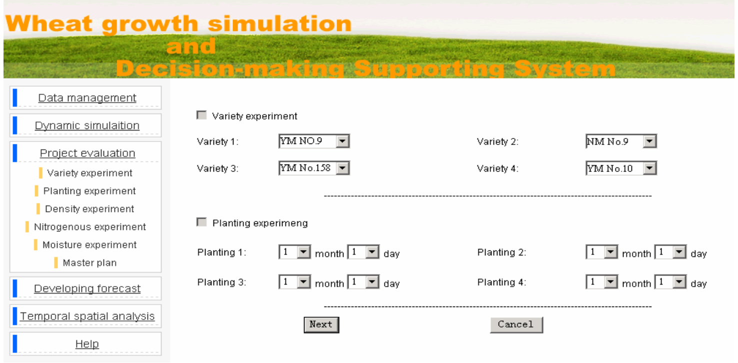
(3) According to instruction, interpreter simulation engine inquiry the location of according business component from component directory to obtain component or instantiated hotspot subsystem. The management glue center would be responsible for glue and plug it upon DAcSA. Then these business component and subsystem would run dynamically to implement required functions.

(4) When agriculture-crops business requirement or simulation model change, RMA model will change. Under the revised or new rules, the SensorInf in the Controller will get the input information and transform the condition (EvolvedInf) to evolve the existing component or produce the hotspot library. Based on RMA model and in light of the new rules, the

adaptability of DAcSA will be implemented through selecting the new structure block from component library and hotspot directory.

## 3. CONCLUSION & DISCUSSION

Research on DAcSA-based architecture adaptability can effectively meet the domain-agriculture-crops business requirement and adapt the system evolution with business requirement change for different simulation model construction and different crops, and improve the productivity of domain software. The proposal of RMA model and virtual machine framework concept is to simplify the domain-agriculture-crop software development, particularly to high efficiently implement the different simulation model or revise model from different domain specialist by software. DAcSA-RMA model has been applied in digital agricultural crops system, which proves preferable adaptability, as shown in Figure5.



□ Variety experiment

Variety 1: YM NO.9 Variety 2: NM No.9

Variety 3: YM No.158 Variety 4: YM No.10

□ Planting experimeng

Planting 1: 1 month 1 day Planting 2: 1 month 1 day

Planting 3: 1 month 1 day Planting 4: 1 month 1 day

*Figure 5.* Wheat growth simulation & Decision-making Supporting System

The simulation model is different with different crops and different agricultural specialist. DAcSA-RMA is put forward to extract the common factors between them, implement the smooth transformation and flexible adaptability from business demand to software implementation. However, more research would be still necessary for different agricultural crops application system. Mainly they are:

DAcSA uses unchangeable black-box component and changeable white-box hotspot subsystem, i.e. DAcSA of the black and white box mixed framework to solve the constructing block. Evidently, change and fixedness are relative. When business requirement changes, the black and white box mixed constructing block will transform and evolve, which requires timely increase and update of component library and hotspot knowledge library. Therefore, the establishment and management of component and hotspot subsystem library must keep abreast of the time and be dynamic.

Extraction of business logic rules and management of rule library is another key. Extraction method of regulations, semantic description of regulations, grammatical structure and management of rule library are all crucial to adaptive model of DAcSA.

Management glue center is the code collection to plug and play upon DAcSA for business component and the hotspot subsystem to be instantiated. Directory service in the virtual machine is use to label the physical location of business components and other components. The common component interface is key technique to achieving seamless connection between constructing blocks, which is subject to further research.

## ACKNOWLEDGEMENT

The research and thesis writing was sponsored by "Research on Precise Agriculture Production Design and Management Decision- making Model Technology" (Item: 2006AA10A303) under the State Hi-tech Development Plan (Plan 863). Our gratitude also goes to Jiangsu Hi-tech Information Agriculture Key Laboratory for their support and assistance.

## REFERENCES

Barbara H R, Karl P, Philippe L, et al. A domain-specific software architecture for adaptive intelligent system [J]. IEEE transactions on Software Engineering. 1995.21(4):288-301

Cao W X, Zhu Y, Tian Y C, Yao X, Liu X J. Research progress and process of digital farming techniques, Scientia Agriculture Sinica, 2006.39(2):281-288.

Guo Zh, Zhao X B, He F, Gu M, Sun J G, Software oriented to special domain model and architecture for enterprise information system, Computer Integrated Manufacturing System, 2004.9:1046-1051.

Huang S X, Fan Y Sh, Zhao Y. Research on generic adaptive software architecture style, Journal of software, 2006.6:1338-1348

Kwozacznski W. Achitecture framework for business components. 5th Internantional Conference on Software Reuse [C]. Computer Society Press, 1998.300-307

Li X R, Xu H L. Design and Implementation of Business Component Oriented Black & White Box Domain Framework, Mini-micro system, 2005.1: 64-68.

Li Y, Wu ZH H, Research on architecture of dynamic self-healing system, Journal of Zhejiang University (Engineering Science), 2005.2:216-222.

Mei H, Shen J R. Progress of research on software architecture, Journal of software, 2006.6:1257-1275.

Shang J G, Liu X G, Hua W H. Urban geological domain specific software architecture, Earth Science-Journal of China University of Geoscience, 2006.9:673-677.

Xu H L, Li X R, Business component reengineering based on rule library, Computer Integrated Manufacturing System, 2003.10:911-915.

Yan M C, Cao W X, Luo W H, Jiang H D. A mechanistic model of phasic and phenological development in wheat [I]: Assumption and Description of the Model. Chinese Journal of Applied Ecology, 2000.11(3):355-359.

Yu ZH H, Cao Y L, A study on the domain specific software architecture, Micro-electronic and computer, 2004.21(7):66-69.