Toward Core Point Evolution using Water Ripple Model

Zhibing Yu, Kun Ma*
Shandong Provincial Key Laboratory of Network Based Intelligent Computing
University of Jinan
No.336, West Road of Nan Xinzhuang, Jinan, 250022, Shandong
CHINA
zhibingyu@yeah.net, *Corresponding author:ise_mak@ujn.edu.cn

Abstract: With the continually-increasing capability of computer hardware and scale of computer software, the complexity of software is also continually increasing, and has manifest itself as one of the key factors that limit the significant improvement of software quality and productivity. However, traditional methods, such as waterfall-like model, component-based programming, software architecture method and agent oriented programming, have encountered more fundamental difficulties in processing complex system development. To address this challenge, we have witnessed that the thought of water ripple model origins from the evolution of water wave when a stone is threw into the water. In this paper, we have proposed three designing core point models, including the high level abstract core point model, the feature core point model and the function core point model, and interactive mode with each other aiming to control the structural complexity of the whole software life cycle. Furthermore, we have proposed a water ripple model of software development process with loosely-coupled correlated core point evolution. The development of a complex system is translated into the evolution of core points. With the persistent evolution thought of water ripple, the development process has strong expansibility and flexible reusability.

Key–Words: software development process; water ripple model; core point evolution; software architecture; agent oriented programming; component-based programming

1 Introduction

To develop software in software industry, the software development team follows a strategy that consists of processes methods and tools.Over the past four decades, we have witnessed numerous studies and significant improvement in the field of software development. The publication that Winston Royce proposed now famous waterfall model [1] in 1970 heralded the beginning of a new era for software development, an era which recognized that software development is a complex collaborative team activity that is subject to many pitfalls if not appropriately organized. Based on the imperfections of the perfect model of the waterfall including the demand change, development risk and iterative development, the researchers put forward a series of waterfall variant model such as spiral model, prototype model[2],iterative model and so on. At the very beginning, these models and their variants were accepted widely. However, developers failed to manage large and complex projects [3] [4].

Motivated by need for open architectures that continuously and automatically evolve new components, Agent Oriented Programming (AOP) was a new agent oriented software development methodology. For instance, the explicit representation and manipulation of goals and plans facilitates a run-time adaptation of system behavior in order to cope with unforeseen circumstances, or for a more meaningful interaction with other human and software agents [5]. The component-based software reuse and development is considered as an effective and efficient approach to improve the productivity and quality of software development, and is applied widely in building distributed systems. With the continuous growth of the software, component need a strong ability to adapt the external environment. For example, it need a strong interaction ability and evolution ability. After the analysis of the roadmap of software architecture (SA), the researches of software architecture focus on not only the design phase, but also covers every phase of software life cycle [6].

In order to solve the above limitation from another prospect, this paper have proposed a water ripple model of software development process based on core point. The core of this method is core point evolution. Figure 1 shows the landscape of water ripple model. We have witnessed that the thought of water ripple model origins from the evolution of water wave when a stone is threw into the water. Similarly, the collaborative software development process can be managed in such a way. Compared with traditional software de-

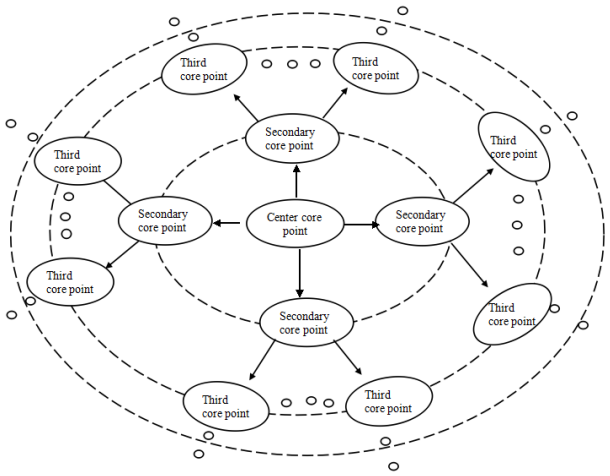

Figure 1: Landscape of water ripple model

velopment process, customers become the important role of a core decision-making group in a team. Frequent requirements of customers are taken fully into account. During the development process, resource allocators mainly engaged in the design of each iteration as well as the development of core point, which greatly shortens the development cycle. Afterwards, the assemblage of the core points as well as their evolution can deliver a running software product in a controlled way. As depicted of Figure 1, different core points are not absolutely isolated. Each core point has the characters of high cohesion and loosely coupled. With the persistent evolution thought of water ripple, the development process has strong expansibility and flexible reusability.

The contribution of this paper are several folds. First, The business of the project is divided into different stages by using the high level abstract core point model, the feature core point model and the function core point model. Second, high level abstract model to the implementation of low level function can be running evolution by the water ripple thought. Third, The strongly coupled of core points increases to the interior of parent core point ,thus reducing the coupling between the core points. Finally, the role of core decision-making group enables investors, developers and users work together to fulfill dynamic demand.

The remainder of the paper is organized as follows. The related work is discussed in Section 2. Section 3 describes the design of three models and interaction model with each other. Section 4 describes in detail the implementation of water ripple model, including definition of roles, and development of central and subsequent core point, and evolution of core point. Section 5 gives a case study of core point evolution using an online shopping mall(OSM) system. Section 6 deeply compare with other methods in order to show the advantages of this method. Brief conclusions and future work are outlined in the last section.

2 Related Work

2.1 Traditional Software Development Method

Over the past four decades, we have witnessed numerous studies and significant in the software development process, such as waterfall model [1], prototyping model [2], and spiral model . At the very beginning, these models and their variants were accepted widely, However, developers failed to manage large and complex projects [3] [4].

2.2 Emerging Software Development Method

Over the latest decade, some emerging development methods have been discussed. Software architecture [6], component-based programming [7], agent oriented programming[5] and model-driven engineering are four typical examples.

2.2.1 Software Architecture Software Development Method

As an important means to control the complexity of software systems, to improve software quality and to support software development and software reuse, software architecture [6]has become a mainstream research field in the software engineering community.The importance of stressing the components and their connectors of a software system is generally recognized and has lead to better control over the design, development and evolution of large and increasingly dynamic software systems. Early studies focused on the design phase of the software life cycle, and generated a new occupation that was called software architect. Although software architecture method has remarkable achievements in controlling software complexity, it has great limitations on other software development stages.The structural complexity, especially the control the high-level structural complexity, is considered as the most important complexity of a software system. How to control the high-level structural complexity in efficient and effective ways has become a critical issue for large-scale software development in the Internet computing environment, which is open, dynamic and constantly changing[5][8]. To resolve this problem, people extends the thinking of software architecture from software design into the whole soft-

ware life cycle.The main methods can be concluded as follows:

- Several researchers have proposed that a conceptual view can describe basics and main structure of system architecture about the whole software life cycle [5].

- Some students have put forward the integration method of SA life cycle.

- The ABC(Architecture-Based Component Composition) [8]was called an SA centric software development method. It unifies the core artifacts in different software life cycle phase into different kinds of SA models, and then the core activities of software development,deployment, maintenance and evolution are performed as the continual refinement,mapping, and transformation to these different kings of SA models .

Deep compared with water ripple software development process, we can see the sixth about deep comparison between methods.

2.2.2 Component-based Software Development Method

Component-based software reuse is considered an important and feasible approach to solving software crisis, and the component model which depicts the intrinsical features and the composition is one of the essential ingredients for realizing reuse[9][10]. The definition of component interface can be divided into to the type of structure level and behavior level.Be based on structure level, CORBA[11] component is not the direct use of other components for provide service, but use the foreign request service of interface definition in order to reduce coupling degree and reveal dependence between components. The behavior level is based on the structure level and adds the dependence and service order of relations between the various services. For example,Reo uses the abstract behavior type to define the behavior of the component on the port[12]. The thought of the method of combination that is called connector separates the control and calculate in order to greatly reduce the coupling degree and reuse by component. Component model based on Wright and literature [13] [14] are a kind of component model.

Deep compared with water ripple software development process, we can see the sixth about deep comparison between methods.

2.2.3 Agent-Oriented Software Development Method

Agent-Oriented programming(AOP) is inspired from the concepts and metaphors of multi-agent systems and borrows agent theory and technology to construct software systems. To qualify for an agent, a software or hardware system is often required about properties such as autonomy, social ability, reactivity, and productivity . Although agent oriented software engineering has achieved many remarkable results, it faces many problems . These problems is mainly concluded in the following aspects:

- As is known to all, requirement is the starting point of a project , and also is the main power of promoting the development of technology. Due to the point, researchers have taken steps to understand this requirement. For example ,there were several typical solutions. The requirement of system is presented by the concept of objectives, roles, actor, agent. In order to distinct between hard and soft targets, i^* have proposed strength for modeling about no functional requirement. Tropos distinguished between early requirement analysis and late requirement analysis in order to understand the demand of user[15]. In this paper, we combine above these two methods and introduce the concept of roles, objectives, core point in order to present the requirement of system. In addition, this paper adopts a hierarchical water ripple evolutionary thought that controls the right evolution of software by the strategy of the requirement decomposition evolution .

- Efficiency, cost and quality are key factors for software engineering. However, many experts research on AOP from the viewpoint of artificial intelligence, and the results obtained are only focused on some specific technical aspects [16]. Therefore, it leads to rarely widely accepted in industry. This paper very sensitive to these aspects and adopt a series of strategies. The paper uses the parallel development and design iteration for efficiency. In addition, the role of resource allocator is responsible to for making reasonable plans to shorten the development cycle by the analysis to the current situation. And the role of component integrator designs reusable components of particular core points for a long-term development of the same industry. The paper makes a series of measures for software quality. For example, the paper provides a protocol verification for the same level of core points and a submitted verification for the staged evolution.

And the paper can use object-oriented verified technology for a single core point. Additionally, each stage of software evolution can be strictly controlled. Meanwhile, the role of core decision-making group that enables investors, developers and users work together make sure the consistency of demand and evolution.It is well known that the best way to reduce cost is to improve the efficiency of development and the quality of product. Besides, the core decision-making verify and accepts for the stage of product, thus is the grade of evolution in order to control cost.

2.2.4 Model-driven Software Development Method

The last popular development method is model-driven engineering (MDE) [17] [18]. With this approach, final system is generated by model transformation from platform-independent model (PIM) to platform-specific model (PSM). This approach emphasizes the abstraction of the business, which is similar to our water ripple model. But the difference is that our abstraction is core point evolution.

3 Based on Business of Core Point Model

The traditional component mainly focuses on the function, and system is completely dependent on the combination of the functional components. The difficulty is very large. In order to reduce the complexity of software, this paper proposes that the core business is the source point of the project, and adopts the high level abstract core point model, the feature core point model and the function core point model to implement the water ripple development. Thought the level evolution of water ripple, the business of system implements the mapping of high level abstract to the feature area until the business function display of low layer. This paper will introduce the definition of the core point of water ripple, the design of three core point models and interactive mode between core points.

3.1 Definition of Core Point

Definition 1 Predicate $INSTANCE\ Y = INSTANCE(X)$ means that Y is an instance of X. That is to say that X is meta of Y.

Definition 2 Core point CP It is basic data type of an element, denoted as CP.

Definition 3 Core point set cps Core point set is a series of core points, denoted as $cps = \{cp|cp = INSTANCE(CP)\}$.

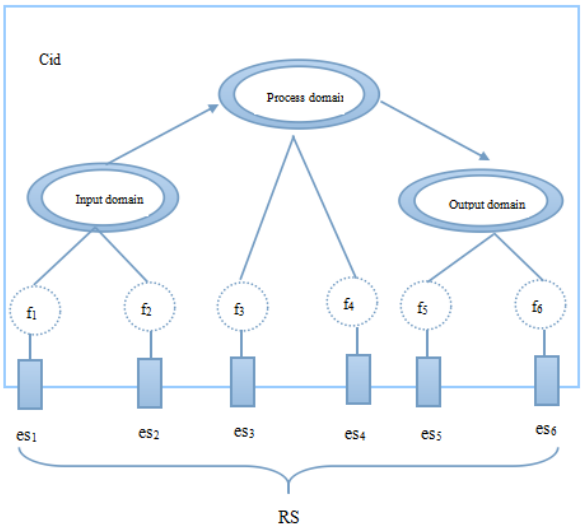

Figure 2: The high level abstract core point model

Definition 4 Projection operator i_th Projection operator i_th is the ith element of set $s = \{a_i|a_i = INSTANCE(CP)\}$.

Definition 5 Central core point cp_{11} It is a unique central core point of determined project, denoted as $cp_{11} = 1\_th(cps)$.

Definition 6 Secondary core point cp_{2i} Secondary core point cp_{2i} $(min <= i <= max)$ is the evolution of cp_{11}. During the process of core point evolution, the number of core points has a clear scope about every level. Generally, the bound of scope is presented by min and max. And this scope will be discussed on the following.

Definition 7 N-level core point cp_{ni} N-level core point cp_{ni} is evolved from its upper-level core point $cp_{(n-1)j}$, where $1 <= i <= n$, and $1 <= j <= n$. Therefore, $cps = \{cp_{11}, cp_{21}, ..., cp_{2n}, ..., cp_{i1}, ..., cp_{ij}, ...\}$.

3.2 Modeling of Core Point

Development of each project, the development team and investors are very clear what is the development core business. But through depth analysis of demand, we can find the corresponding entity and map to complex business logic with each other, thus carrying out the design, encoding and testing. This tight coupling, once the demand for changes or product upgrades, will cause greater development costs, and even the end of the project. How to adopt a strategy is to reply on the high level of abstraction of the business and the low layer to implement function similar the assembly of components. And in accordance with a mechanism, we can transform high level abstract business into the implementation of low level function. This high

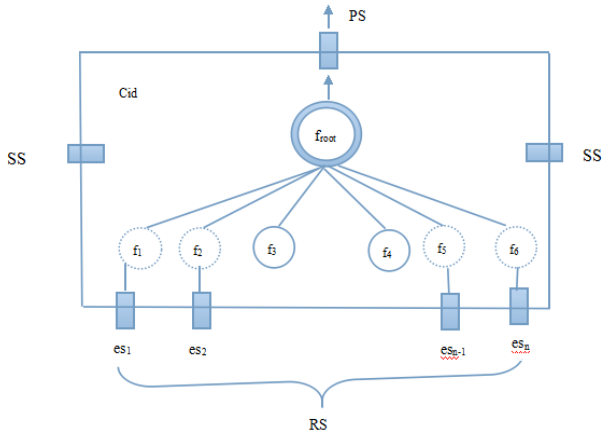

Figure 3: The feature core point model

level dependent abstraction is well decoupled, while the low level combination of the component speeds up the efficiency of the development and saves cost. In addition, according to the mechanism, we can find the demand point of change and carry out the demand change in order to meet the requirement of users at any time. In this paper, we will design the high level abstract core point model, the feature core point model and the function core point model.

3.2.1 The High Level Abstract Core Point Model

The core business of a project is clear.At the same time it is also the source of various complex business. Core business must be designed to be very simple, and do not involve the implementation of specific functions. To add new requirements or to change the demand, the high level will do not change. This is what is said by the high dependence on the abstract and not dependent on the specific. Based on the core business, the paper divides the abstract business into the characteristics requirement of the input domain, process domain and output domain. The feature demand will map the corresponding evolutionary interface.

Definition 8 The high level abstraction core point model: This model describes the core business of project as the central core point of water ripple model . The abstract core point model can be express $C < cid, f_{abstract}, F, RS >$. Cid is the unique identifier of core point. $f_{abstract}$ is a set of abstract domain, called $f_{abstract} = \{f_{input}, f_{process}, f_{output}\}$. F is the feature set of domain decomposition, called $F = \{f_1, f_2, f_3, ..., f_{n-1}, f_n\}$. $RS = \{es_1, es_2, ..., es_n\}$(requirement service set) are requirement service set for core point. Figure 2 shows the model of core point in the central part of the water ripple model .

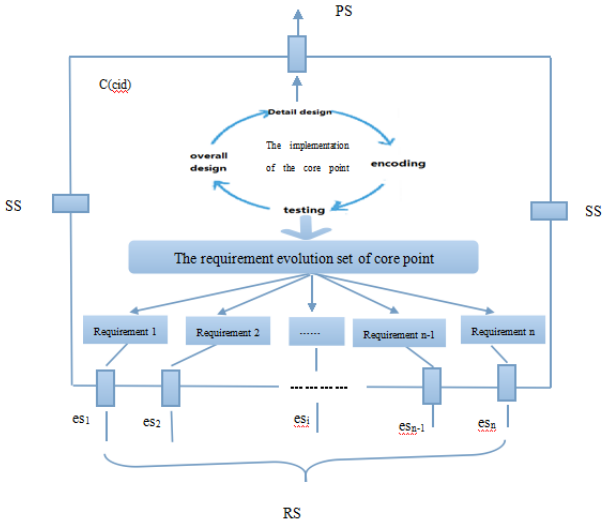

Figure 4: The function core point model

3.2.2 The Feature Core Point Model

Definition 9 The feature core point model: The feature core point model represents the feature subspace for a particular business area of a project. The feature core point can be expressed $C < cid, f_{root}, F, PS, RS, SS >$. Cid is the unique identifier of core point. f_{root} is the root feature of feature space of feature core point. F is the subcharacter set of feature decomposition, called $F = \{f_1, f_2, f_3, ..., f_{n-1}, f_n\}$. SS(share interface set) are the sharing information interface set for the same of level core point. PS(provide service set) is providing service set for core point. RS(require service set) are requirement service set for core point.Figure 3 shows the model of core point.

3.2.3 The Function Core Point Model

Definition 10 The function core point model : The function core point model represents the core business extract of a project and these auxiliary business maps the corresponding interface. Then it can form a running project through the combination and evolution. Core point model can be expressed $C < cid, f_{entity}, SS, PS, RS >$. Cid is the unique identifier of core point. f_{entity} is the realization body of core point. SS(share interface set) are the sharing information interface set for the same of level core point. PS(providing service set) is providing service set for core point. RS(requirement service set) are requirement service set for core point. Figure 4 shows the model of core point.

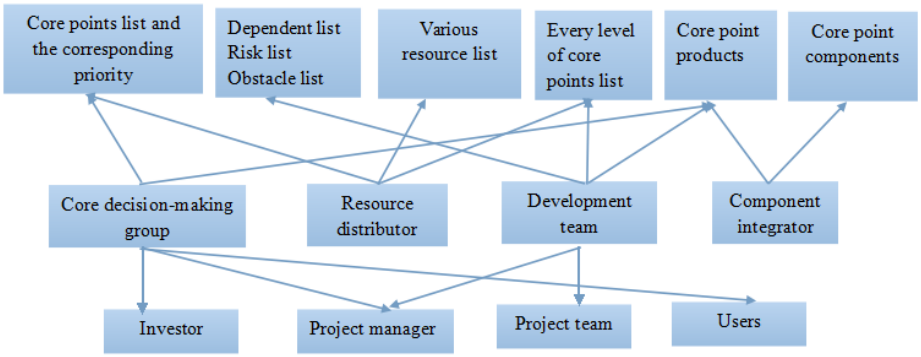

Figure 5: The role relation graph of water ripple software development process

3.3 Interactive Mode of Core Points

The different core points implement a integrated project by interactive way. This paper will introduce three kinds of interactive modes, which are combination, dependency and correlation. The combination relationship is caused by the relationship between the demand of characteristics and the supply of characteristics. Dependency is caused by a number of core points in a certain order. And the correlation relationship is caused by the characteristic decomposition. $C_1 < f_{root}, F, SS, PS, RS >$ and $C_2 < f_{root}, F, SS, PS, RS >$ belong to the same feature space Ω.

1. if $\exists$ $f_1 \epsilon$ PS(C_1),$\exists$ $f_2 \epsilon$ PS(C_2),and $f_1 = f_2$,there is a combination of the relationship between C_1 and C_2 ,expressed as $C_1 \xrightarrow{f1-f2} C_2$.
2. if $\exists$ X $\subseteq$ F(C_1), $\exists$ Y $\subseteq$ F(C_2),and $X \rightarrow Y$ is founded in the Ω, there is a dependency relationship between C_1 and C_2 , expressed as $C_1 \xrightarrow{f1 \rightarrow f2} C_2$.
3. if $\exists$ X $\subseteq$ F(C_1), $\exists$ Y $\subseteq$ F(C_2), the P(X$\cup$Y)is a semantic feature of Ω, and there is correlation relationship between C_1 and C_2 ,expressed as $C_1 \xrightarrow{f1 \cup f2} C_2$.

4 Implementation of Water Ripple Model

Water ripple model is a new method of software development process based on the evolution of core point. In a highly collaborative development environment, the research and development (R&D) team include investors, developers, users and other about roles.The role of core decision-making group is responsible for defining an integrated core point for customers. Once the core point is determined, and the development team will accomplish a series of tasks including design, coding, testing. Afterwards, the core decision-making group continues to improve the product in the iterative stage. Next, the core decision-making group continues to design subsequent core point to fulfill the basic requirement of customers.Therefore, the objective of the development is providing satisfactory delivery to customers. In this process, all members of the team are collaborated together. This paper will introduce the definition of roles of water ripple model, the development of central core point, evolution modeling of core point and the development of secondary core point.

4.1 Roles of Water Ripple Model

1. Core decision-making group: This role enables core point evolution in a dynamic environment. It is responsible for the entire project plan, and determine the differences of core points including demands and priority. That is to say that core decision-making group is the key to the project. Generally, core decision-making group might be a customer, project manager, developer and end user.
2. Resource allocator: the role makes the reasonable plan of resource allocation by the analysis of current situation. Generally, resource allocator is a developer.
3. Development team: This role collaborates with the core decision-making group to make and evolve core points. Generally, development team is composed of several developers.
4. Component integrator: This role designs many components for the core point modules that are stored in the component library and are reused in the future.

The relations of different roles, resources, products and practical application roles are show in Figure 5.

4.2 Development of Central Core Point

4.2.1 Determination of Central Core Point

The central core point is determined by the core decision-making group. Some factors (technology, risk, and fund) are considered before the determination of central core point. First, the R&D team comprehends the demand of the customers to determine central core point. For example, the central core point is logistics or booking in the logistics management system or booking management system.

4.2.2 Development of Central Core Point

Central core point has the following restraints.

1. The implementation of project should be simple and logic clear.
2. The development life cycle is not more than one month.
3. The framework is open.

Since central core point is clean among all the core points, waterfall is a reasonable model to develop and implement the central core point. The team turns to design, code, and test the system after the core decision-making group determines the requirements of customer.

4.3 Evolution Modeling of Core Point

4.3.1 Definition of Core Point Evolution

Core point is the basic element in the process of core point evolution. Subsequent core points are evolved using evolution rule from the central core point. Therefore, core point evolution is expressed as three tuple, denoted as $cpe :< A, \Sigma, R >$, where

1. **Definition 8 Evolution Site:** It is a series of function interfaces about the evolution of core points.The core decision-making group makes a series of function sets that are defined by the requirement list and are sorted by priority, in order to evolve subsequent core points. A is a set of evolution sites of cpe, denoted as $A = \{es_i | es_i \ \ is \ \ an \ \ evolution \ \ site\}$, where es_i is the evolution place of core point.
2. **Definition 9 Evolution Action:** It is a evolution strategy from a core point to another core point. Evolution action adopt a reasonable evolution strategy in order to improve the quality of products by the external environment perception,current resources(personnel, development schedule, cost) and software performance require. In the water ripple model, we present five basic evolution action: multiple evolution (ME), disjoined evolution (DE), complete evolution (CE), incomplete evolution (IE), and user-defined evolution (UE). Since the evolution action from one core point to another is time-related, the evolution action is changed with the requirements. Σ is a set of evolution action of cpe, denoted as $\Sigma = \{\sigma_i | \sigma_i \ \ is \ \ an \ \ evolution \ \ action\}$.
3. R is a set of evolution action rules of cpe.It demonstrates what kind of actions should be performed to achieve high-level evolution strategy by the core decision-making group.

Next, we analyze the evolution actions further.

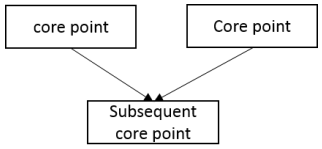

Figure 6: Multiple evolution.

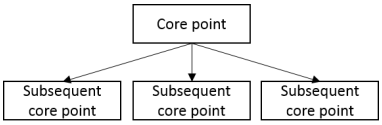

Figure 7: Disjoined evolution.

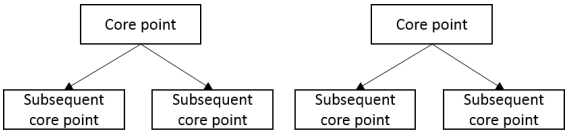

Figure 8: Complete evolution.

1. Multiple evolution ME: The evolution of core points implements the mapping from parents to one child. Figure 6 shows multiple evolution.
2. Disjoined evolution DE: The evolution of core points realizes the mapping from unique one parent to one child. Figure 7 shows the disjoined evolution.

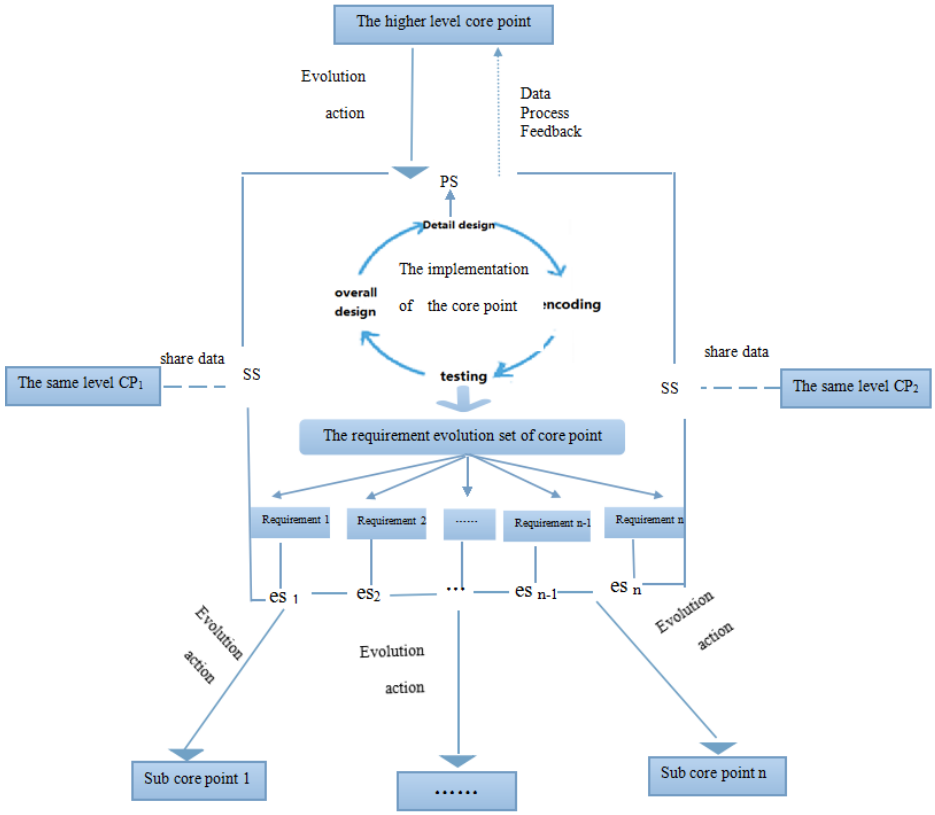

Figure 10: The evolution modeling

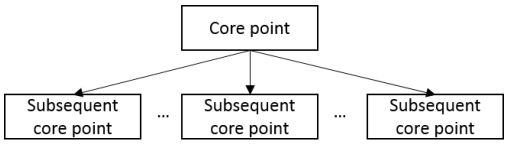

Figure 9: Incomplete evolution.

3. Complete evolution CE: The evolution of core points achieves the mapping where the child has fixed siblings. Figure 8 shows the complete evolution.

4. Incomplete evolution IE: The evolution of core points accomplishes the mapping where the child has alterable siblings. Figure 9 shows the incomplete evolution.

5. User-defined evolution UE: The evolution of core points implements the mapping that is not the above four evolution.

4.3.2 Evolution Mechanism and Evolution Modeling of Core Point

Cp_{11} is the highest abstract entity and the global goal for the whole project as the central core point.The evolution of Cp_{11} as follows:

Internal condition: Requirementlist()$\bigwedge$ Evolutionlist()

Select $es_i \in \{es_1, es_2, ..., es_n\}$

External condition: Arrangement()$\bigwedge$Development()$\bigwedge$Design()

Select $\sigma \in (ME, DE, CE, IE, UE)$

The central core point can evolve many sub core points. And these sub core points cooperate with each other in order to make sure the complete operation of the whole project.This paper can introduce a form about the running of core point evolution.

Core point

{

Globalgoal The higher level core point

Localgoal (initial condition1):function1(I, O)

(initial condition2):function2(I, O)

.....

Coordination-rules Cp_{11} (O=Cp_{ij}_evolution)

{

Core_decision-making_group.Requirementlist();

//the core decision-making defined Cp_{11} list of requirements

Core_decision-making_group.Evolutionlist();

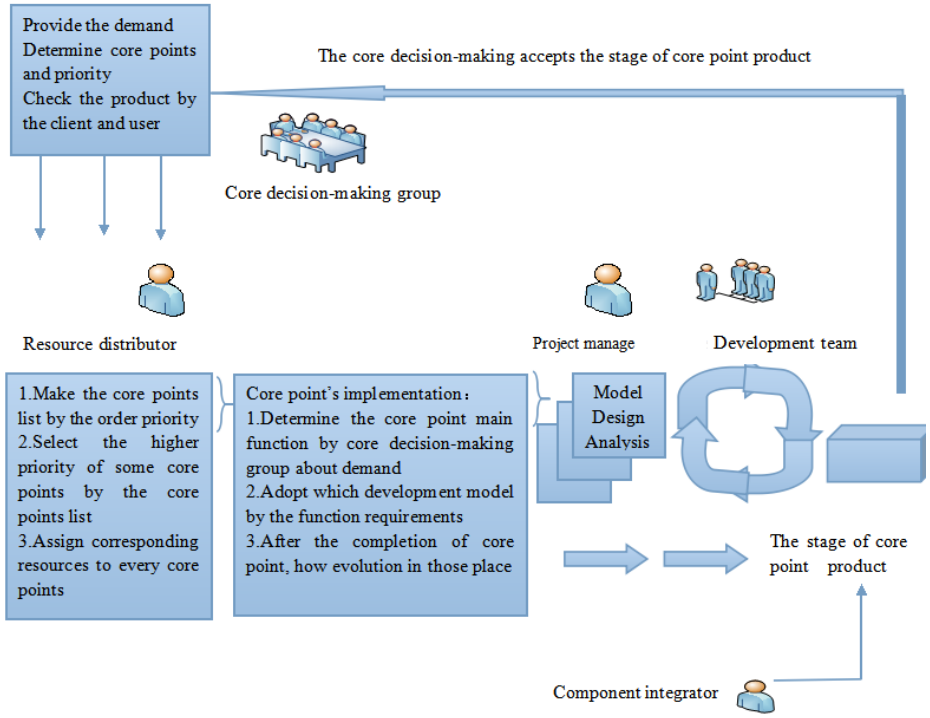

Figure 11: Development of second core points

//the core decision-making defined Cp_{11} a series of evolution sites

Resource_allocator.Arrangement();

//resource allocator allocates kinds of resources to each sub core point by their priority Development_team.Development();

//the development team develops the core point

Component _integrator.Design();

//component integrator designs component for his core point that are stored in the component library in order to reused in the future.

}

}

Globalgoal means that the father of the core point can achieve objective.Through the decomposition of globalgoal, the core point should accomplish these tasks meaning localboal. Localgoal contains initial condition, function, precondition and postcondition, where I denotes preconditions and O denotes postconditions. Coordination-rules means a rules that a variety of roles can work well together.

Evolution sites and actions of central core points generate subsequent core points. The core decision-making group defines the requirement and corresponding priority. Project managers organize a meeting to discuss how to implement these core points, including the select of development model and method. Next, waterfall model is used to complete the development to deliver a product to customers. The siblings of core points have weak relations to interchange the data. Afterwards, evolution sites and actions of subsequent core points are evolved. Figure 10 shows the evolution of core points.

4.4 Development Process of Core Point

After the determination of central core points by the core decision-making group, resource allocators quantize the priority of core point. The core decision-making group provides the results of requirement analysis. Afterwards, the R&D team selects the specific development model. For example, waterfall model is appropriated in the context of explicit requirements, and prototyping model is appropriated in the context of high-risk requirements. Next, the R&D team delivers the testing product to the customers. Finally, feedbacks from the customers are used to help developers to start another iterative development. Figure 11 shows the development of second core points.

The standard process of water ripple model is in the following.

1. The core decision-making group defines a series of core points that are given the corresponding priority.

2. The resource allocator distributes kinds of resources to each core point with current state and resource.

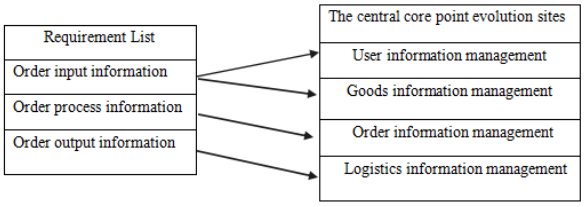

Figure 12: Requirement evolution

3. The development team routinely develops the core point under the resource arrangement, such as selection of models, system design, detailed design, coding, and testing.

4. The developed stage product is delivered to the core decision-making group who will review the feedback of core points to improve the product.

5. The team retrospects the core points in a meeting to summarize the issues and reused experiences

6. Component integrator can design component for the core point module that are stored in the component library and are reused in the future.

7. Next core point is made and evolved in cycles before the delivery of final product.

5 Case Study

5.1 Case Overview

We take an online shopping mall(OSM) as an example of core point evolution. The OSM consists of four parts: user management, order processing center(OPC), goods management and logistics management. The order is generated and is sent to the OPC. And OPC can generate shopping request and deliver to administrators. Afterwards, the administrator approve the order and deduct the payment of users. Otherwise, the order can be rejected. Next, the processed orders are sent to the logistics, which packs up the products in stock and ships to the customer. Another invoice is generated at the same time.

5.2 Core Point Evolution

To development such an an online shopping mall(OSM), we adopt water ripple thought as the development mode. Next, we will describe the core point evolution. The core decision-making group have an absolute leadership during software development process. For example, it continuously validates the customer

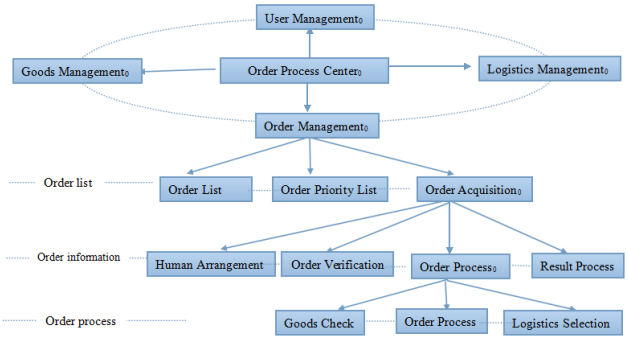

Figure 13: Core point evolution

requirements and feedbacks of intermediate products to absolutely control the development process. The continuous evolution emphasizes the stage evolution from simple to complex.

Definition cp_0**:** It presents the high-level abstraction of core point, and includes own function target,the precondition and the restraint of processing results.

5.2.1 The internal function evolution of high-level core point

The core function of OSM is order processing, which is the central core point of water ripple model. The core decision-making makes the demand analysis for the OPC_0, and produces a series of demand set. And the core decision-making defines a set of evolution sites$(es_1, es_2, ..., es_n)$ and the corresponding priority. The core decision-making selects a group of the highest priority of evolution sites in order to evolve some secondary core points.According to the scope management idea, OPC_0 can evolve to the secondary core points by the internal and external conditions.It includes four high-level abstractions of secondary core points:user management,goods management, order management and logistics management. The internal evolution grape from demand to the high-level abstract entities as follows figure 12 :

5.2.2 The external performance evolution of core points with each other

According to these evolution sites,the resource allocates the corresponding resources.The development team integrates these information and develops the running stage product. Afterwards, the product is delivered to the core decision-making group to be confined, and then feedback are provided to improve the product until reaching the standard. During the evolution process, the evolution action typically adopts disjoined evolution and complete evolution.While the

demand of project will change or the online of project will continue to evolve about some core points, the evolution action adopts incomplete evolution. Due to space limitations, we introduce the evolution chains of order list, order management, and order processing. Afterward, the order management evolves into order list, order priority list and order acquisition. Next, we take the evolution of order acquisition for example. The order acquisition evolves into human arrangement, order verification, order process, and result process. Next, the order process evolves into goods check,order process and logistics selection. Specific core point evolution diagram as follows: OPC_0 is the highest level of abstraction entity and includes its own function target, the precondition and the restraint of processing results. It reflects OPC_0 as a higher level environment view in the whole project.The global goal of OPC_0 is composed of user management, goods management, order management and logistics management.OPC_0 and the above relations consist of a high level of organization mode and they cooperate work in order to complete the project. Figure 13 shows the evolution process of OSM.

6 Deep Comparison Between Methods

6.1 Comparison with Component Model

Compared with the traditional component model, the water ripple model has the following advantages:

6.1.1 Core Business Extraction

The design of component model is rarely involved in the business domain, and it is not to say that it is studied in different levels of business. This paper focuses on the different business areas from the core business to the characteristics of the various areas of business until the realization of the functions. we can design three core point models, which is the high level abstract core point model, the feature core point model and the function core point model. The complex business will become very clear and be reduced in a large extent.

6.1.2 The Reducing Complexity of Water Ripple Model

The traditional component-oriented development is based on the framework of the functional combination and communication by the connector. There is not a mechanism to focus on the interaction between different levels of business and the business of the same level.In this paper, we have proposed a water ripple model of software development process with loosely-couple correlated core point evolution. This method can realize the mapping from high level abstract model to the implementation of low level function. The water ripple model perfectly realizes the control complexity of system.

6.1.3 Loose Coupling Based on High Level Interaction

This paper mainly introduces that the strongly coupled of core points increases to the interior of parent core point ,thus reducing the coupling between the core points. According to the feature extraction, water ripple reducing complexity and decoupling of core points, these realize the evolution modeling from the abstract core point to feature core point. It can be seen that the feature abstract domain of the central core point is $f_{abstract} = \{f_{input}, f_{process}, f_{output}\}$. The abstract domain is divided into input domain(f_{input}), process domain($f_{process}$), output domain(f_{output}). And the three domains are extracted from the abstract feature and map to the input feature f_1 and f_2 ,process feature f_3 and output feature f_4. f_1 and f_2 is the semantic relationship of the input domain ,expressed as $f_1 \cup f_2$, as the dependent input of f3 (expressed as $f_1 \cup f_2 \rightarrow f_3$). f_3 and f_4 are similarly dependent relationship.

From figure 14, we can see that the C_1, C_2, C_3, C_4 of the same level core points are transformed into the implicit association of C_0 core point. At the same time, the shared information interface is used to achieve the weak relation between the core points of the same level in order to strengthen interaction. The information shared of the same level core point and the logical relationship between the interaction with core points from parent core point are very good to achieve the separation of data and business.

6.2 Comparison with software architecture

Compared with software architecture, the water ripple model has the following advantages:

6.2.1 Dual Control of Development and Management

Compared with software architecture development process,water ripple model will adopt two levels control about development and management. In the development level,this paper uses a independent running core point as a small project for the decomposition of

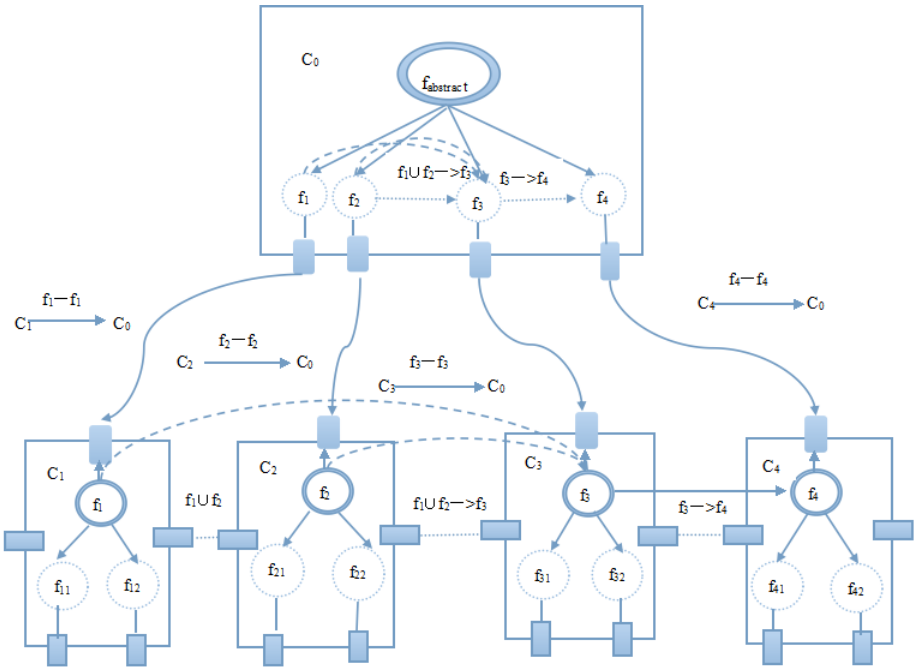

Figure 14: mode of second core points

project. Additionally, these core points can coordinate running as the evolution way of water ripple from central core point to the periphery. In the management level, this paper sets a new conception of core decision-making group as the central hub of the whole project. On the one hand, it as a central hub can mobilise kinds of roles in order to coordinate running. On the other hand, it can control the correct direction of evolution for the project as the demand of customer.

6.2.2 The Control of Local and Global Complexity

Software architecture is focused on the control of high level complexity. In this paper, we can focus the control of local area complexity and the control of global complexity of the project. In the control of local complexity of different levels, this paper designs three core point models, which is the high level abstract core point model, the feature core point model and the function core point model. The complex business becomes very clear and reduces its complexity in a large extent. High level abstract model to the implementation of low level function can be running evolution by the water ripple thought in order to the control of global complexity of the project.

7 Conclusion

To address the limitations of software architecture methods, component and agent, we have proposed a water ripple model of software development process. The core of this method is core point evolution that combines engineering thought, persistent evolution and component method together. We also present three core point models and evolution modeling. An case study using water ripple model illustrates the feasibility of core point evolution. Future work is targeted in three directions. First, the core point reuse and evolution performance make the corresponding evaluation criteria. Second, the method can be applied to software development by theoretical proof. Third, The prototype system will be developed using water ripple thought.

Acknowledgements: This work was supported by the National Key Technology R&D Program (2012BAF12B07), the Shandong Provincial Natural Science Foundation (ZR2014FQ029 and ZR2015JL025), the Shandong Provincial Key R&D Program (2015GGX106007), the Doctoral Fund of University of Jinan (XBS1237), the National Training Program of Innovation and Entrepreneurship for Undergraduates (201410427030), and the National Natural Science Foundation of China (61573166 and 61572230).

References:

[1] Winston W Royce. Managing the development of large software systems. In *proceedings of*

IEEE WESCON, volume 26. Los Angeles, 1970.

[2] Walter R Bischofberger and Gustav Pomberger. *Prototyping-oriented software development: Concepts and tools*. Springer Publishing Company, Incorporated, 2011.

[3] Mary Shaw. Continuing prospects for an engineering discipline of software. *Software, IEEE*, 26(6):64–67, 2009.

[4] Linda Northrop, Peter Feiler, Richard P Gabriel, John Goodenough, Rick Linger, Tom Longstaff, Rick Kazman, Mark Klein, Douglas Schmidt, Kevin Sullivan, et al. Ultra-large-scale systems: The software challenge of the future. Technical report, DTIC Document, 2006.

[5] M Hadi Valipour, Bavar AmirZafari, Kh Niki Maleki, and Negin Daneshpour. A brief survey of software architecture concepts and service oriented architecture. In *Computer Science and Information Technology, 2009. ICCSIT 2009. 2nd IEEE International Conference on*, pages 34–38. IEEE, 2009.

[6] Hong Mei and Jun-Rong Shen. Progress of research on software architecture. *Ruan Jian Xue Bao(Journal of Software)*, 17(6):1257–1275, 2006.

[7] Ivica Crnkovic, Michel Chaudron, and Stig Larsson. Component-based development process and component lifecycle. In *Software Engineering Advances, International Conference on*, pages 44–44. IEEE, 2006.

[8] Hong Mei, Gang Huang, and Tao Xie. Internetware: A software paradigm for internet computing. *Computer*, (6):26–31, 2012.

[9] Roberto Di Cosmo and Jr00me Vouillon. On software component co-installability. *Acm Transactions on Software Engineering Methodology*, 22(4):256–266, 2011.

[10] Bo Chen, L. I. Zhou-Jun, and Huo Wang Chen. Research on component models:a survey. *Computer Engineering Science*, 30(1):105–109, 2008.

[11] M. A. Mei-Juan. Design and development of the virtual experiment system based on technology of corba and java. *Research Exploration in Laboratory*, 2012.

[12] Farhad Arbab. Abstract behavior types: a foundation model for components and their composition. *Science of Computer Programming*, 55(1):3–52, 2005.

[13] Kung-Kiu Lau, Perla Velasco Elizondo, and Zheng Wang. Exogenous connectors for software components. In *Component-Based Software Engineering*, pages 90–106. Springer, 2005.

[14] Bo Chen, ZhouJun Li, and HuoWang Chen. A new component-oriented programming language with the first-class connector. In *Modular Programming Languages*, pages 271–286. Springer, 2006.

[15] Michalis Pavlidis and Shareeful Islam. Sectro: A case tool for modelling security in requirements engineering using secure tropos. *Caise Forum*, pages 89–96, 2011.

[16] MAO Xin-jun. State-of-the-art, challenges and perspectives of agent-oriented software engineering [j]. *Computer Science*, 1:004, 2011.

[17] Kun Ma, Bo Yang, Zhenxiang Chen, and Ajith Abraham. A relational approach to model transformation with qvt relations supporting model synchronization. *Journal of Universal Computer Science*, 17(13):1863–1883, 2011.

[18] Kun Ma, Bo Yang, and Ajith Abraham. A template-based model transformation approach for deriving multi-tenant saas applications. *Acta Polytechnica Hungarica*, 9(2):25–41, 2012.