



Journal of
**Software
Engineering**

ISSN 1819-4311



Academic
Journals Inc.

www.academicjournals.com

Dynamic Modeling Self-Adaptive Software by Intelligent Service Components

¹R. Fan, ¹Y.Q. Peng, ¹Y.F. Chen and ²X.H. Liu

¹School of Software, Guangdong Ocean University, China

²Department of Finance, Guangdong Ocean University, China

Corresponding Author: R. Fan, School of Software, Guangdong Ocean University, China

ABSTRACT

In order to construct independent autonomous, dynamic evolved, flexible software entity and dynamically assemble them under the internet for self-adaptive software, novel intelligent component model is critical. This paper propose to fusion agent, service and component for building autonomy, dynamic evolved, intelligent accumulated, great granularity Intelligent Service Component (ISC), provide the good software entity for self-adaptive software. We try to expand all functions into services, on command dynamic integrate relative services to Intelligent Service System (ISS). A case shows that the Intelligent Service Component (ISC) is able to dynamic rebuild and intelligent interact each other for software system adaptability. A novel self-adaptive software system is constructed by intelligent service components.

Key words: Self-adaptive software, dynamic modeling, intelligent service components

INTRODUCTION

Self-adaptive is activities to actively adjust its behavior when testing environment change and its state at run time, for software to ensure sustained, high quality services (Ding *et al.*, 2013). Adaptive software can assess their own behavior, when the evaluation shows its own is not close to the tendency to accomplish the goal or it may provide better functionality or performance, software change their behavior (Laddaga, 1999). Ding *et al.* (2013) reviews the existing research and practices based on the “Sensing decision-execution” basic self-adaptation cycle. It present the concept of software self-adaptation and taxonomy of the characteristics of sensing, decision and execution activities, summarizes those most focused enabling techniques for constructing adaptive software, analyzes a set of typical research projects and then describes the future research trend in the field of constructing adaptive software. Kakousis *et al.* (2010) aims to provide a disambiguation of the adaptation, as it is understood in ubiquitous computing and a critical evaluation of existing software adaptation approaches. It focus on existing solutions that enable dynamic software modifications that happen on resource constrained devices, deployed in mobile and ubiquitous computing environments and puts forward its development trends and challenges. Macias-Escriva *et al.* (2013) compares Component Based Software Development, Aspect-Oriented Software Development and Agent Based Software Development. The purpose of this comparative analysis is to give a succinct and clear review of these three methodologies. Their definitions, characteristics, advantages and disadvantages are considered and a conceptual mind map is generated that sets out a foundation to assist in the formulation and design of a possible new integrated software development approach. There are four aspects like the self-adaptive principle

and the conceptual framework, self-adaptive system modeling and model evolution mechanism, runtime system self-adaptive adjustment mechanism, social systems software self-adaptive, etc., which give, special (Jin and Lv, 2013; Zhang and Zhao, 2013; Liu, 2013; Mao and Sun, 2013; Peng *et al.*, 2013) answer why software systems need to have self-adaptive ability and how have. Previous studies (Ding *et al.*, 2011; He, 2010; Chen and Li, 2010; Liu *et al.*, 2011; Wang *et al.*, 2013; He *et al.*, 2008; Shi *et al.*, 2014) exploded that self-adaptive software models from different angles.

MATERIALS AND METHODS

In order to construct Independent autonomous, dynamic evolved, flexible software entity and dynamically assemble them under the internet for self-adaptive software system, novel intelligent component model is critical. The most active research on self-adaptive software entities are component, service and agent. Component assembles software system by providing external interfaces. Service builds larger, loose component by service composition. By perceiving, reasoning and optimizing, agent actively responds to external changes. We fusion agent, service and component to build autonomy, dynamic evolved, intelligent accumulated, great granularity Intelligent Service Component (ISC) and use services to implement leaf sub-goal which is decomposed by goal tree, on command dynamic integrate relative services to establish a novel self-adaptive software system.

Intelligent service component model: The component entity concerned how to flexibly add, delete or replace components for adapting to changes; the service entity concerned how to dynamic integrate heterogeneous, loose coupling, larger granularity services; agent interested in perception, reasoning and autonomous behavior. They alone can't solve contradiction conflict due to software continual evolution for adapting changes of environment or user new demands.

We propose to integrate advantages of component, service and agent, expand agent into the Intelligent Service Component (ISC). Implement interactive of components with environmental by sensors and effectors, implement intelligent reasoning by BDI reasoning center, achieve the capacity of dynamic evolution of components by dynamic service composition machine, with request and provides interface to realize the integration between intelligent service components. Intelligent service component model is as shown in Fig. 1.

Self-adaptive ring of ISC: The differences between ISC with BDI agent are described as below:

Capability: Environment perceiving+mental reasoning+service assembling+interface interacting.

Reasoning chain: Environment→sensors→events→beliefs→goals→services→plans→effects→environment.

Self-evolving mechanism: On demand generate goal tree, actively find and assemble services to achieve leaves sub-goal.

Components integrate mechanism: On demand discover bind services and integrate ISCs by service interfaces.

Self-adaptive ring of intelligent service component is as shown in Fig. 2.

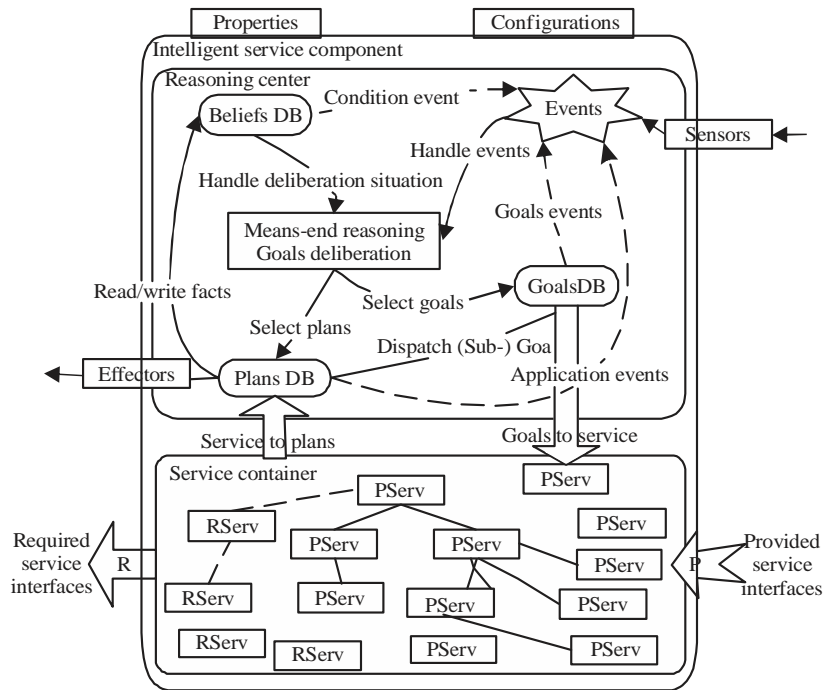


Fig. 1: Intelligent services component model

Perception forms goals by extenics reasoning: Intelligent control center perceives the changes of environment or user demand by sensors and form events queue. Inference is executed based on the nature of events and the current state of mind (Mentality situation) of beliefs (Beliefs DB) for semantic of change demand by analysis event semantics. By extenics rhombus reasoning, functional requirements and non-functional requirements goal tree is formed and is written to the goals (Goals DB). Changes of environment or user need may make beliefs change and produces new goals to the goals (Goals DB). New inference generates new goal tree for replying to change of internal event (Fig. 1 dashed arrow). The internal state of beliefs, goals and plans change is self-adaptive ability to adapt to various changes.

Active services aggregate to realize leaf sub goals: The service container is gathering center of services. On demand find services which implement leaf sub goal and actively aggregate services into corresponding service tree (Goals to service). Each service on the service tree, may already has a Provide Service (PServ) or must require an external service (RServ) from other intelligent service component. Intelligent service component on demand dynamic evolution is only two core actions: increase a service node or remove a service node, other actions are implemented by a combination of actions (for example, service node is replacement by remove the old service node and add a new service node). Increase service is also two cases: One case is by purchasing and downloading external services to the local as to service node (PServ). Another is by requesting external services return results to local as to service node (RServ and dotted line connection) and with service complete these nodes disappear. With provided services and request service aggregation mechanism, multiple intelligence service components synergy to accomplish more complex

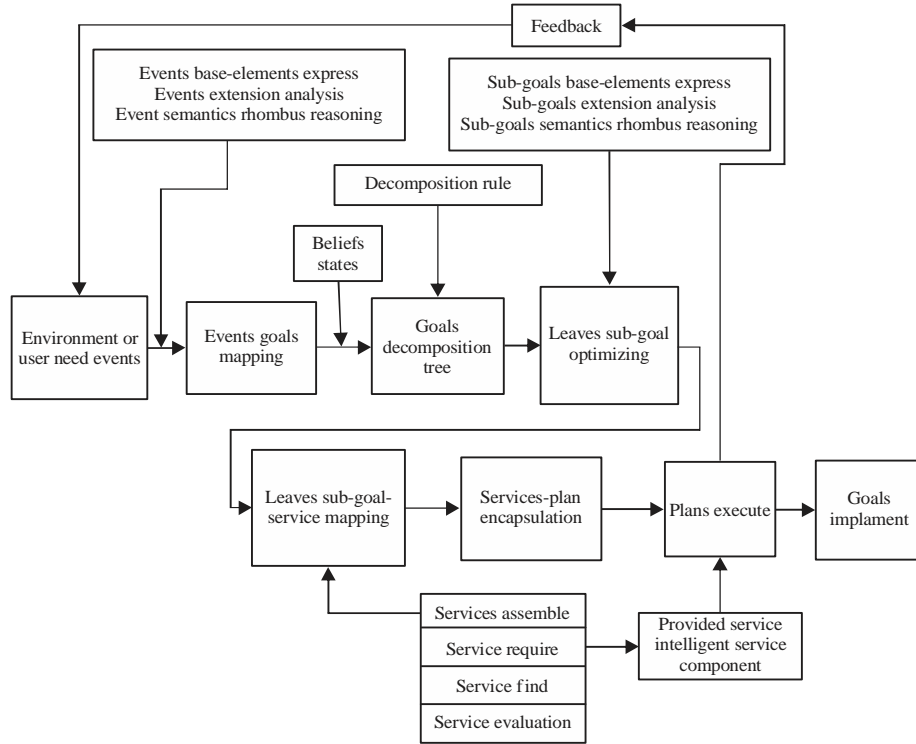


Fig. 2: Self-adaptation ring of intelligent service component

goals. Buy permanent service or rental temporary service for realizing intelligent service system dynamic evolution depends on the economy and the importance of balance.

Package services to plan: By encapsulating service tree into plan (Service to plans), plans (Plan DB) is updated and is executed in turn by effector to further lead to new changes of environment or user demand.

In order to improve dynamic adaptive ability, intelligence service component also provides custom property (Properties) and system configuration function. To tune characteristics of intelligent service component at runtimes, user can get and set property values. To realize dynamic evolution of intelligent service component by rewriting XML format configuration file.

Integrating ISCs to self-adaptive software: By intelligent service component, we can ease integrate relative Intelligent Service Component (ISC) to form Intelligent Service System (ISS) which run on internet. On demand dynamic integrating mechanism make intelligent service system to have self-adaptive capability. Self-adaptive software model is as shown in Fig. 3.

In the diagram, an intelligent service component request service interface is connected to other intelligence services component provide service interface with a dotted line. This expresses intelligent service components can dynamic connect each other on demand through a pair request and provide service interface and collaborate to finish more complex tasks. An enterprise software system is built by some intelligent service components which are dynamic rebuild and control on demand.

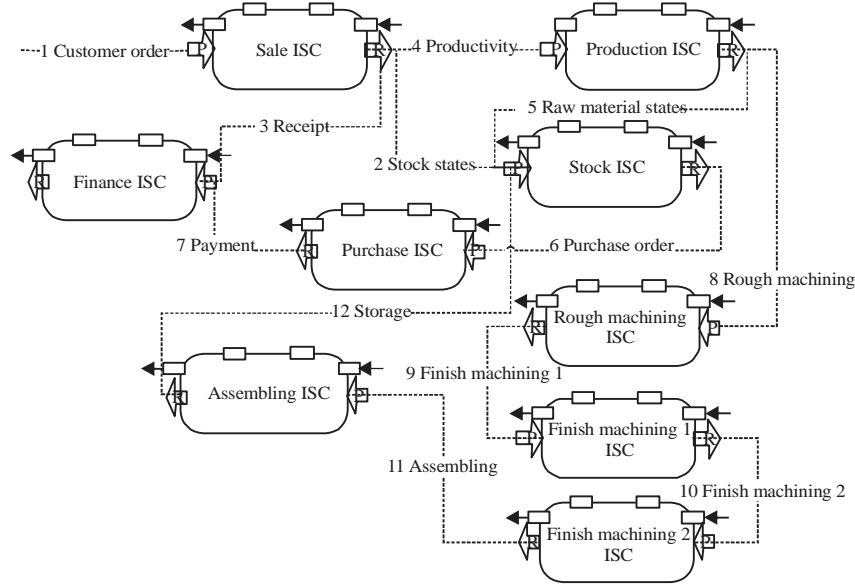


Fig. 3: Self-adaptive software system

Dynamic rebuilding: Its dynamic rebuilding analysis is as follows:

```

Configure {
NodeSet (node1,node2,...,nodek,...,nodem);/*Internet nodes set*/
ISCSet (SaleISC,PurchaseISC, StockISC, ProductionISC,FinanceISC, RoughMachiningISC, FinishMachining1ISC, FinishMachining2ISC,
AssemblingISC);/*intelligent service components(ISCs) set*/
Conf(node1, SaleISC);/* sale ISC in node 1 */
Conf(node2,StockISC);/*stock ISC in node2 */
Conf(node3,PurchaseISC);/* purchase ISC          in node3*/
Conf(node4,ProductionISC);/*production ISC in node4 */
Conf(node5,FinanceISC);/*finance ISC in node5 */
Conf(node6,RoughMachiningISC); /*RoughMachining1 ISC in node6 */
Conf(node7,FinishMachining1ISC); /*FinishMachining1 ISC in node7 */
Conf(node8,FinishMachining2ISC); /*FinishMachining2 ISC in node8 */
Conf(node9,AssemblingISC);/*Assembling ISC in node9 */
}
    
```

Intelligent Interacting: Its intelligent interacting analysis is as follows:

- When customer sends out order service request to sale intelligent service component (Sale ISC), sale ISC carries out customer crediting. If crediting is failed, customer order is refused
- If customer is right, sale ISC sends out stock states service request to stock intelligent service component (Stock ISC) for goods amount. Stock ISC queries stock and returns results to sale ISC
- If goods amount is larger than order amount, sale ISC informs customer payment and sends out receipt request to finance intelligent service component (Finance ISC) for payment for goods. When finance ISC returns OK result, sale ISC delivers goods to customer
- If goods amount is smaller than order amount, sale ISC sends out productivity service request to production intelligent service component (Production ISC) for production capability. If capability is insufficient, customer order is refused

- If production capability is sufficient, production ISC sends out raw material states service request to stock ISC for raw material amount. Stock ISC queries stock and result to production ISC
- If raw material amount is insufficient, stock ISC sends out purchase service request to purchase intelligent service component (Purchase ISC) for purchasing raw material
- Purchase ISC informs supplier to deliver raw material. After raw material ordered have been checked and accepted, purchase ISC sends out payment service request to finance intelligent service component (Finance ISC) for paying to supplier
- When raw material is sufficient, production ISC sends out rough machining service request to rough machining intelligent service component (Rough machining ISC) for first machining procedure based on machining plan
- Rough machining ISC sends out finish machining 1 service request to finish machining 1 intelligent service component (Finish machining 1 ISC) for second machining procedure
- Finish machining 1 ISC sends out finish machining 2 service request to finish machining 2 intelligent service component (Finish machining 2 ISC) for third machining procedure
- Finish machining 2 ISC sends out assembling service request to assembling intelligent service component (Assembling ISC) for final machining procedure
- Assembling ISC sends out storage service request to stock ISC for storing production to stock

Likewise, dynamically composes sale ISC, stock ISC and finance ISC to further complete delivers goods task.

With dynamic change of user needs and business flows are constantly emerging, only need to find or develop new services and packaging to intelligence service component and provide the service results to other request service caller, we can dynamically reconfigure new software system which has self-adaptability.

RESULTS

For providing the good software entity to dynamic integrate self-adaptive software system, we fusion advantage of agent, service and component to build autonomy, dynamic evolved, intelligent accumulated, great granularity Intelligent Service Component (ISC) model. We use services to implement leaf sub-goal which is decomposed by ISC's goal tree, on command dynamic integrate local or remote services to establish a novel self-adaptive software model. A case shows that the Intelligent Service Component (ISC) is able to dynamic rebuild and intelligent interact each other for software adaptability.

DISCUSSION

The differences between our findings of study with previously published studies are described as below:

The ISC's capability is stronger than some published studies (Ding *et al.*, 2013; He *et al.*, 2008). It includes environment or user need changes is perceived by sensors or request service interfaces, goal is formed and decomposed by extenics rhombus reasoning, leaf sub-goal is achieved by services assembling and ISC entities are integrated by service interfaces.

The ISC's reasoning chain is difference with the cognitive agent (Shi *et al.*, 2014) and other BDI agent (Macias-Escriva *et al.*, 2013; Kakousis *et al.*, 2010), service entities are core for dynamic adapting changes by adding, deleting, modifying or aggregating service entities:

Environment→sensors→events→beliefs→goals→services→plans→effects→environment

The ISC's self-evolving mechanism is difference with other studies (Chen and Li, 2010; Ding *et al.*, 2011; Liu, 2013), which is implemented by on demand generating goal tree, actively finding and assembling services entities to achieve leaves sub-goal.

The ISC's integrate mechanism is difference with some published studies (Liu *et al.*, 2011; He, 2010; He *et al.*, 2008; Ding *et al.*, 2011; Shi *et al.*, 2014; Macias-Escriva *et al.*, 2013; Kakousis *et al.*, 2010), which is achieved by on demand discovering, dynamic binding and integrate intelligent service component which has provide service interfaces on local or remote nodes.

CONCLUSION

This research proposes to fusion agent, service and component for building autonomy, dynamic evolved, intelligent accumulated, great granularity Intelligent Service Component (ISC), provides the good software entity for integrating self-adaptive software. By on demand generating goal tree, dynamic assembling services and so on self-evolving mechanism, intelligent service component can dynamic adapts changes of environment or user need. By dynamic integrating intelligent service components by service interfaces, make software system has self-adaptive capability.

Further work is needed to develop customized intelligent service component tools and integrating platform for improving capability of dynamic constructing self-adaptive software system.

ACKNOWLEDGMENT

This research is supported by the scientific and technological projects of Guangdong province, China under Contract 2014A040402010.

REFERENCES

- Chen, H.L. and R.F. Li, 2010. Survey on self-adaptive evolution software. *Applic. Res. Comput.*, 27: 3612-3621.
- Ding, B., H.M. Wang, D.X. Shi and X. Li, 2011. Component model supporting trustworthiness-oriented software evolution. *J. Software*, 22: 17-27.
- Ding, B., H.M. Wang and D.X. Shi, 2013. Constructing software with self-adaptability. *J. Software*, 24: 1981-2000.
- He, K., R. Peng, W. Liu, J. Wang and B. Li, 2008. *Networked Software*. Science Press, Beijing, China.
- He, K., 2010. The research progress for software method and standardization based on on-demand services. *Commun. CCF*, 6: 21-25.
- Jin, Z. and J. Lv, 2013. Why and how to achieve self-adaptive ability for software systems. *Commun. CCF*, 9: 1-12.
- Kakousis, K., N. Paspallis and G.A. Papadopoulos, 2010. A survey of software adaptation in mobile and ubiquitous computing. *Enterprise Inform. Syst.*, 4: 355-389.
- Laddaga, R., 1999. Creating robust software through self-adaptation. *IEEE Intell. Syst. Applic.*, 14: 26-29.
- Liu, Z., X. Zhu and B. Cao, 2011. Method of adaptive programming. *Comput. Eng. Applic.*, 47: 80-82.
- Liu, L., 2013. Adaptation and evolution requirements of software systems services. *Commun. CCF*, 9: 13-20.

- Macias-Escriba, F.D., R. Haber, R. del Toro and V. Hernandez, 2013. Self-adaptive systems: A survey of current approaches, research challenges and applications. *Expert Syst. Applic.*, 40: 7267-7279.
- Mao, X. and Y. Sun, 2013. Self-adaptation technology for social technical system. *Commun. CCF*, 9: 20-26.
- Peng, X., W. Zhao and B. Chen, 2013. Requirement-driven software self-adaptation. *Commun. CCF*, 9: 27-36.
- Shi, Z., J. Zhang, J. Yue and X. Yang, 2014. A cognitive model for multi-agent collaboration. *Int. J. Intell. Sci.*, 4: 1-6.
- Wang, Z.D., H.Q. Wang, G.S. Feng, H.W. Lv and X.M. Chen, 2013. Research on autonomic computing system and its key technologies. *Comput. Sci.*, 40: 15-18.
- Zhang, W. and H. Zhao, 2013. Some thinking on trustworthiness and self-adaptability of software. *Commun. CCF*, 9: 8-13.