Software Component Reusability Metrics

특집
05

목 차

김영철 · 김우열 · 서윤숙 · 김기두
(홍익대학교)

1. Introduction

In the current software industry it is necessary to develop software products with high reliability and quality. To do it, most developers should be focusing on interoperability, reusability, and composability for developing new system development. Hudgins[1] said, "Interoperability has the characteristic of a suite of independently-developed components, applications, or systems that implies that they can work together, as part of some business process, to achieve the goals defined by a user or users. Reusability has the characteristic of a given component, application, or system that implies that it can be used in arrangements, configurations, or in system-of-systems beyond those for which it was originally designed. Composability has the ability to rapidly assemble, initialize, test, and execute a system from members of a pool of reusable, interoperable elements. Composability can occur at any scale—reusable components can be combined to create an application, reusable applications can be combined to create a system, and reusable systems can be combined to create a system-of-systems."

In this paper, we will only focus on reusability. And briefly mention component based software engineering to identify and extract software components for reusability.

Component Based Software Engineering (CBSE) is one way to produce software products, which assembles and reuses the existing component pieces[2]. However, CBSE still has its problems such as the interface and size of components, configuration management, version control, etc. In CBD(Component Based Development) some methods (UML component method or feature modeling) of domain analysis only identify components within the particular system domain and not allude to the importance/frequency of identified components. A component may actually include the common characteristics (functions or applications) of the system family. Feature oriented domain analysis (FODA) is focused on bottom up approach, which identifies many kinds of features during analysis and then identifies certain components through the commonality and variability of features [2,5,6,7,8,12]. Feature is defined as a prominent or distinctive user visible aspect, quality, or characteristics of a software system or systems. However, this approach in immature domains or large systems may have a lot of disadvantages: complexity of feature modeling, extraction of meaningless features, etc. Our workflow oriented domain analysis (WODA) is focused on analyzing the static and behavior of systems or applications through top down approach. This mechanism incrementally and iteratively identifies diverse components from high level components to low level ones, depending on what each Planner, modeler, developer, or tester needs.

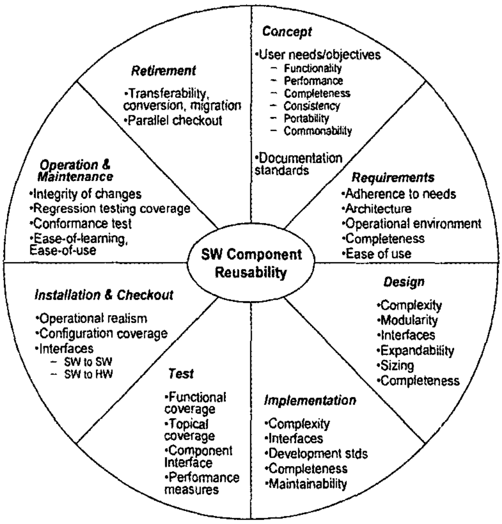
Now, we will carefully consider a matter in all its reusability to develop new system enhanced. First, let's consider how to find reusability of the common/uncommon components of the existing system. Second, let's consider the frequency and the criticality of reusable components. we will mention to enhances productivity of new systems with high quality of reusable components.

Section 2 describes quality attributes for reusability metrics. Section 3 introduce our workflow oriented domain analysis. Section 4 introduce the component test plan metrics for recognizing the important/frequent component. Section 5 shows tool for reusability measurement. The last section mentions our conclusion.

2. Reusability As a Quality Attribute

We have a deeply interesting in the high quality and reliability of software products. But how can do it? We focus on developing new systems through reusing some existed software components with high quality, and also reducing the whole software development life cycle.

There are many different models for software quality, ISO 9126-1991, IEEE Std 982.2-1988. In (figure 1), we suggest quality factors for reusability with reference in IEEE Std 982.2-1988[14,15].



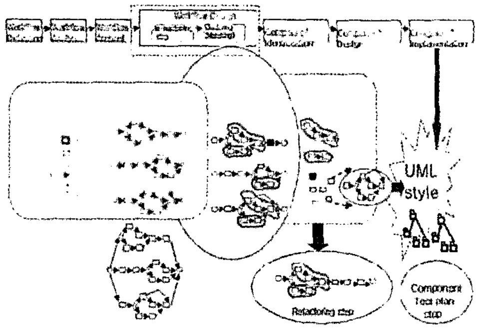
(Fig. 1) Quality Factors Impacting Reusability[15]

Actually building high reusable software depends on the application of quality attributes at each phase of the development life cycle. In focusing on software development for reusability, we need to identify and measure the quality attributes applicable at each life cycle phases. In this paper, we just mention to focus on requirement, design, and test phases. Specially focus on the criticality /frequency of software components at early phases.

3. Workflow Oriented Domain Analysis (WODA)

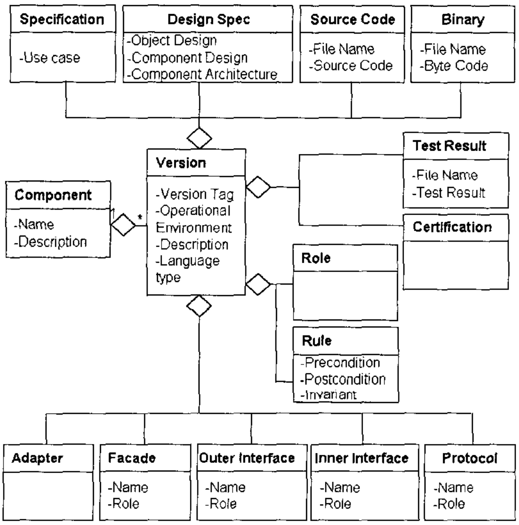
Our WODA focuses on analyzing the static structure and behavior of systems or applications through the top down approach. This mechanism incrementally and iteratively identifies diverse components from high level process components to low level, but isn't mentioned on limited paper.

This process works incrementally and iteratively until the optimal size of components for each one is obtained. (Figure 2) shows a whole procedure of workflow oriented domain analysis. The most reusable component and scenario (path) is identified, especially by the component test plan metrics during the last step.



(Fig. 2) Procedure of Workflow Oriented Domain Analysis (WODA)

(Figure 3) shows the class diagram of our defined component specification. In figure 3, we show our definition of a software component, which be composed of versions. each version is composed of products of designs, codes, test cases, and interfaces.



(Fig. 3) Our meta-model definition of a Component

4. Component Test Plan Metrics

We introduce metrics for component test plan used in the generation of component test plans as part of our workflow oriented domain analysis methodology. The component test plan uses a set of workflows that contain a collection of executable sequences of workflow process model in a particular system domain. Component test plan metrics are employed to enhance the productivity with reusability of the critical and frequent components through analyzing the behavioral scenarios of the whole workflow process model. The purpose of this component test plan metrics is to identify 'reusability' through the criticality / frequency of common/uncommon components in which the scenarios (paths) defined by the rows of the component weighted matrix are executed. This approach was adopted from Musa's work on Operational Profiles [3,4]. Musa's approach assumes that the designer has sufficient insight to assess the 'criticality' of action units and assign weighting factors to the elements of the action matrix [9,10,13]. This approach differs in that the domain analyzer analyzes the scenarios based on the 'reusability' of their components or subpaths.

〈Table 1〉 Component test plan Metrics

		Measures of test path	Weight value (w)
Length	1) Shortest path (simple path) - least steps of actions		$w = 1$
	2) Longest path (hardest path) - most steps of actions		
Criticality	**1) Most critical path**		$w \geq 1$
	2) Least critical path		$w \geq 0$
Reusability	**Component**	1) **Most reusable components**	$w > 1$
		2) Least reusable components	$w \geq 0$ AND ¬=1
	Sub-path	**Most reusable sub-path**	$w > 1$

<Table 1> shows component test plan metrics to measure Criticality and Reusability of particular components. Criticality focuses on the most critical (frequent) component, and reusability on the most reusable components and cluster of components. It also illustrates the component test plan metrics such as most critical scenarios, most reusable components, and most reusable subpaths. First, the issue of <u>*Length*</u> is two aspects of shortest path (i.e., a cluster) and longest path (i.e., a package) for domain analysis. But it is useful if we use this issue with other categories of the metrics. Second, the issue of <u>*Criticality*</u> is important to choose a list of component scenarios (paths). Third, the issue of <u>*Reusability*</u> is also important to identify the reusable components.

To apply test plan metrics for each of the approaches described in figure 3 will be applied to the "Military Integrated Information system" application.

We will calculate total probability of occurrence as follows:

$\forall_i$ *Workflow Scenario* $_i \subseteq$ *A Workflow Model W (W is Military Integrated Information System Application)*

For all workflow scenarios between the starting point and the ending point, the particular workflow Scenario $_i$ is included in a Workflow Model R.

$\forall_i$ *component unit* $_i \subseteq$ *workflow Scenario* $_I$

For all component units within a particular workflow Scenario $_i$ we will calculate the total probability of occurrence with

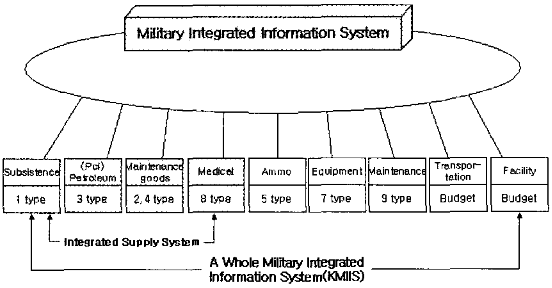
(Π *the weighed factor of component unit* $_i$ * *probability of component unit* $_i$*)* / *(* Σ *probability* $_i$*).*

(See Figure 4 (a),(b),(c),(d))

The Mealy model and the Moore model are theoretically equivalent, but the Mealy model is a link weighted model and the Moore model is a node weighted model[11]. We will apply to both weight concepts. As a result, each component unit is assigned a weighted value with the value one and each link is also a probability of occurrence. But in this paper, We don't mention to calculate total probability of occurrence.

Case Study of Military Integrated Information System

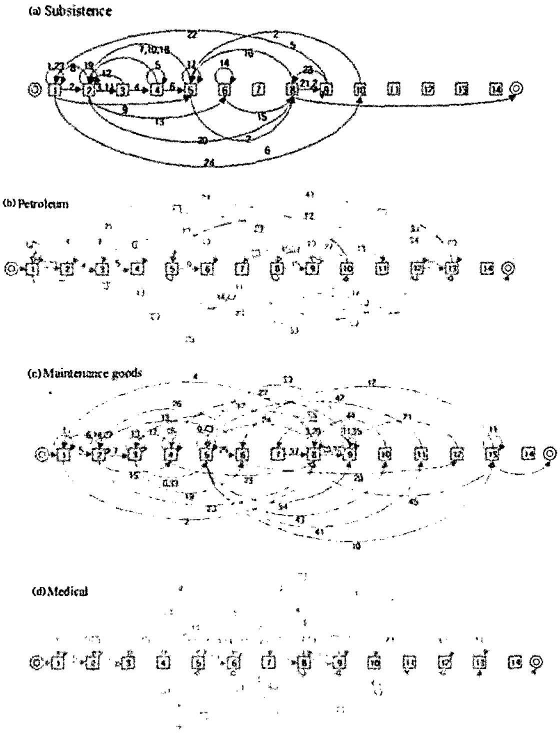
Military integrated information system (MIIS), which is applied to WODA, is based on a huge and complex system[6]. The system consists of 9 sub systems of MIIS: such as Subsistence, Petroleum, Maintenance goods, Medical, Ammo, Equipment, Maintenance, Transportation, and Facility. As a result of analyzing MIIS, each sub system may consist of approximately 14 process components: Catalog specification, Requirement Standards, Funds Responsibility, Plan Budget, Property Management, Transportation Management, Receipt and Payment management, Storage Management, Warehouse Security, Inspection Test, Expend Disposal, Maintenance Management, and Command Valuation. Although it is not possible to show all the steps involved in WODA, figure 4 shows a whole structure of MIIS.



(Fig. 4) Application example of Military Integrated Information System (MIIS)

In this example only four sub systems (rectangles Subsistence, Petroleum, Maintenance goods, Medical) of the whole system will be applied in (Figure 4).

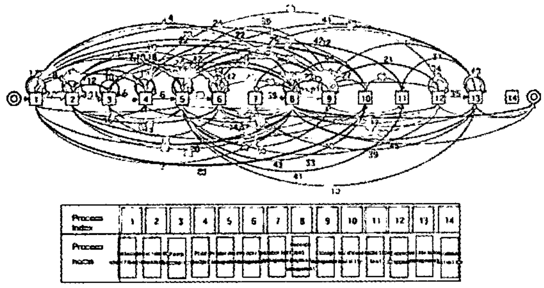
(Figure 5(a)) shows the basic workflow for a general 'Subsistence' service between the starting point and the ending point. (Figure 5 (b)) shows the basic workflow for a general 'Petroleum' service. (Figure 5(c))shows the basic workflow for a general 'Maintenance goods' service. (Figure 5(d)) shows the basic workflow for a general 'Medical' service. In (figure 5), there are 4 sub systems (Subsistence, Petroleum, Maintenance goods, Medical), which are modeled by the WODA model.



(Fig. 5) Workflow Oriented Domain Modeling of Integrated Supply System, sub-systems of Application example of Military Integrated Information System (MIIS)

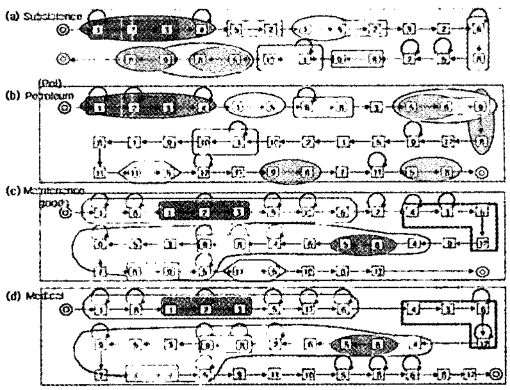
In order to easily recognize common /uncommon components and clusters of components in (figure 7), each sub system is serially arranged and different shapes identify diverse components. These diverse components may be reused when developing other new military integrated information systems. As a result, the time and difficulty of developing new systems are reduced.

(Figure 6) shows to identify the integration of all 4 sub systems of the MIIS. Figure 6 shows criticality (frequency) of reusable components with component test plan metrics in table 1. P1, P2, …,P14 are process components in MIIS. With this metrics, criticality or frequency for the reusable number of the particular component is more clearly identified. Although step 7 of WODA is not mentioned, the identified diverse components from step 6 to step 8 of WODA is easily applied. Easily to explain it, figure 6 shows to identify the diverse components (component, process component, diverse component clusters of components) on integration of each WODA models in (Figure 6).



(Fig. 6) Case study of WODA at Military Integrated Information System (MIIS)

(Figure 6) shows the whole possible workflows of each workflow scenario (path) in the ‘ military integrated information system’ application.



(Fig. 7) Diverse Components within Only 4 sub-systems of MIIS

(Figure 7) shows the various shapes identifying the diverse types of clustered Components for reusability.

4.1 Most Critical Scenario

The first metric is an adaptation of Musa’s ‘most critical operational profile’ approach [3,4]. This metric places greater weight on those workflow scenarios that components should be most critical. It assumes that the designer can make these judgments. Later metrics will not have to assume that someone is available to make such judgments, since they can be produced automatically.

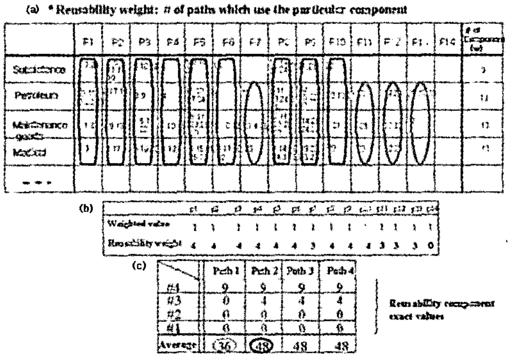
(Figure 7(a)) displays the first direct path of ‘subsistence’ service workflow scenario which consists of the sequence of process components ‘1^*⇒2⇒3⇒4^*⇒5⇒2⇒1⇒5⇒2⇒3⇒2⇒6⇒8⇒5^*⇒2^*⇒8⇒9⇒1^*⇒10⇒5⇒8⇒9⇒8’ with the amounts of weighted values equal to 9.

(Figure 7(b)) displays the second direct path of ‘petroleum’ service workflow scenario which consists of sequences of process components ‘1⇒2⇒3⇒4⇒1⇒5⇒6^*⇒8⇒3⇒5⇒8⇒9⇒8⇒12⇒

9*⇒5⇒1⇒2⇒10⇒1*⇒10⇒9⇒1⇒8⇒13⇒11⇒5 ⇒12*⇒13⇒9⇒8⇒7⇒13*⇒5⇒8' with the amounts of weighted values equal to 13. (Figure 7(c)) displays the third direct path of 'maintenance goods' service scenario which consists of sequence of process components '1*⇒8*⇒1⇒2*⇒3⇒5*⇒13*⇒6⇒2⇒4⇒3⇒6⇒12 ⇒9⇒4⇒8⇒5⇒6⇒2⇒8*⇒9*⇒3⇒5⇒9*⇒7⇒8⇒ 9⇒5*⇒11⇒5⇒10⇒8⇒13' with the amounts of weighted values equal to 13. (Figure 7(d)) displays the fourth direct path of 'Medical' service workflow scenario which consists of sequence of process components '1*⇒8*⇒1⇒2* ⇒3⇒5*⇒13*⇒6*⇒4⇒3⇒6⇒12⇒4⇒8⇒5⇒6⇒ 2*⇒8*⇒9*⇒3⇒5⇒9*⇒7⇒8⇒9⇒5*⇒9⇒11⇒10 ⇒5*⇒8*⇒6*⇒8⇒12' with the amounts of weighted values equal to 13.

It is very hard to apply this MIIS as a huge and complex system with test plan metrics.

(Figure 8(a)) shows the tabular representation of the workflows, component weighted matrix, which apply the calculation of the total frequency of occurrence in each workflow scenarios (paths).



(Fig. 8) Component weighted matrix about Only 4 sub-systems of MIIS

4.2 Most Reusable Components

This approach simply measures the reusability of process components in each row of the component weighted matrix. This matrix places greater weight on those process components that are reused the most by the collective group of scenarios being analyzed. (Figure 8(a)) displays three different types of geometric figures: a rounded rectangle, and an oval. The triangle implies a particular component is used just one time on just a single one of the paths. The rounded rectangle implies that this component us used on four paths.

The oval implies that this component is used on three paths. The reusability weight is defined as the number of paths that use the particular component. Therefore, (Figure 8(b)) shows the values 'reusability weight' of each component. The values can indicate whether a particular component is reusable or not. We may say that the component is reusable when the value of the particular unit is at least 2. (Figure 8(c)) indicates the total values of reusability components on each path (scenario). Due to the 'most critical scenario' and 'most reusable components', We recognize that other paths are more usable than path 1 (subsistence service).

4.3 Most Reusable Subpaths

This metric is similar to the previous metric except that it places greater weight on workflow scenarios which share common subpaths. (Figure 7) shows how to identify each diverse cluster of the sequence of reusable components in all possible scenarios of the military integrated information system application. (Figure 7) also shows various different types of geometric figures: an elliptical figure, a shaded elliptical figure, a diamond figure, an oval and a rounded rectangle for reusability, but it is not important because there may occur many diverse types within this very large and complex application.

On path1 and path2, We can see the 'longest reusable subpath' which is '1' through '4' represented by the ellipse. On path1, path2, path3 and path4, We can see the 'reusable subpath' which is '1' through '3' represented by the rounded rectangle and so on.

The proposed reusability metrics are employed to improve the productivity of component test plan metrics through component scenario prioritization.

5. Tool for Reusability Measurement

From this point we will use another example to explain 'reusability measurement' with the analysis tool. The example is a real time 'Uninterruptible Power System (UPS)' work flow modeling. Focusing on domain analysis view, there are five high level workflow scenarios such as the normal return, the overload, the service interruption, the normal status, and the failure as follows:

5.1 Normal return

when interrupted normal power is supplied to rectifying part and charging part again, battery suspends its discharge automatically, and good quality normal power is supplied to the load without any service interruption through power inverter and at the same time discharged battery is charged again.

5.2 Overload

power inverter automatically synchronizes output frequency, voltage and normal power. When the equipment is out of order or overload, stable power can be supplied to the load under synchronous status with normal power by being switched without any service interruption synchronous switching switch.

5.3 Service interruption

when normal power service is interrupted, the battery, which has charged by rectifying part and charging part in ordinary time, discharges power to supply DC power to power inverter so that the load can supply stable AC power under no power service interruption for specific discharge time.

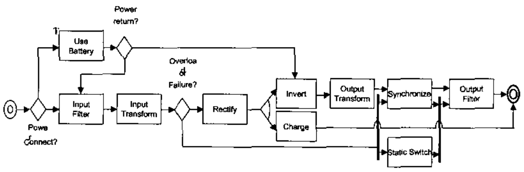
5.4 Normal status

rectifying part and charging part, which receive normal or preliminary power source, shall supply stable AC power by power inverter that switches AC to DC, and shall also charge battery.

5.5 Failure

power inverter automatically synchronizes output frequency, voltage and normal power. When the equipment is out of order or overload, stable power can be supplied to the load under synchronous status with normal power by being switched without any service interruption synchronous switching switch.

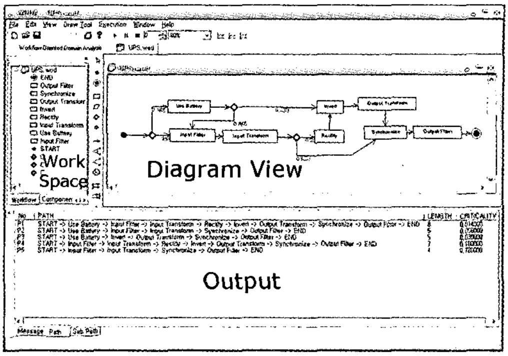
(Figure 9) is based on a real time UPS system, which is applied with workflow oriented domain analysis (WODA) [6]. It consists of 5 workflow paths of UPS such as Normal status, Service interruption, Normal return, Failure, and Overload. As a result of analyzing UPS, each workflow may approximately consist of 8 components: 'Use Battery', 'Input filter', 'Input transform', 'Rectify', 'Invert', 'Out transform', 'Synchronize' and 'Out filter'. We don't show examples to follow all steps of WODA methodology with UPS in this paper.



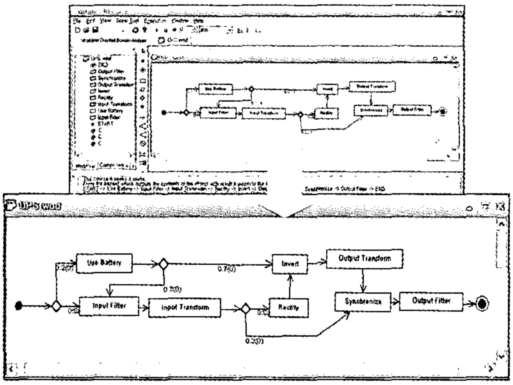
(Fig. 9) Application example of Uninterruptible Power Supply (UPS)

(Figure 10) shows the automatic analysis tool. This tool consists of Work Space, Output View, and Diagram View. Work Space shows the creation of each component. Output view displays the result of extracting components , and scenarios. In the Diagram View, we can draw dynamic component model for the particular UPS. After modeling in this View, just click 'the ▶ executing button' to display all possible component scenarios (paths) in Output view, and to automatically simulate all these scenarios in Diagram view.

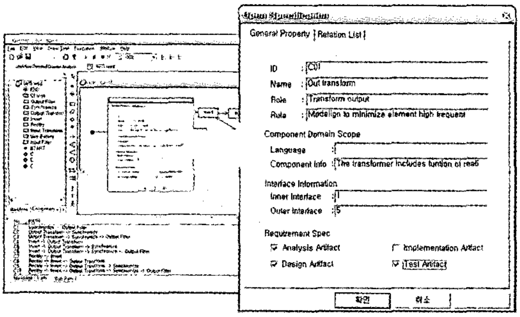
(Figure 11) shows to model UPS workflow. As a result of modeling UPS, each workflow may approximately consist of 8 components: 'Use Battery', 'Input filter', 'Input transform', 'Rectify', 'Invert', 'Charge', 'Out transform', 'Synchronize', and 'Out filter'.



(Fig. 10) An automatic analysis tool



(Fig. 11) UPS workflow modeling



General Property | Relation List

ID : C01
Name : Out transform
Role : Transform output
Rule : Modelign to minimize element high frequent

Component Domain Scope
Language :
Component Info : The transformer includes funtion of reali

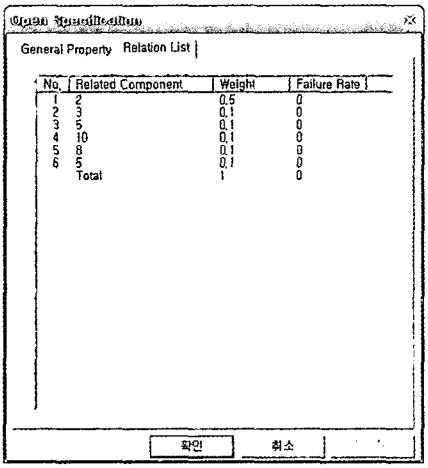
Interface Information
Inner Interface : 1
Outer Interface : 5

Requirement Spec
☑ Analysis Artifact
☐ Implementation Artifact
☑ Design Artifact
☑ Test Artifact

(Fig. 12) A detail example of One component, 'Out Transform' of UPS components

(Figure 12) shows a detail content within 'Out Transform' component of a whole UPS modeling. Each component is contained with the information of component's identification, name, role, rule, interfaces for requirement specification. Each component ID has a unique index value. Each component name help to understand what kind of component is. Role indicates what kinds of role the component have. Rule means that the component keeps the constraints.

The (figure 13) shows to input the weight values of components.

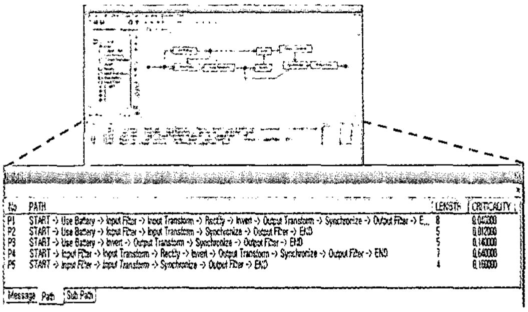


No.	Related Component	Weight	Failure Rate
1	2	0.5	0
2	3	0.1	0
3	5	0.1	0
4	10	0.1	0
5	8	0.1	0
6	5	0.1	0
	Total	1	0

(Fig. 13) The assignments with each weight value of components

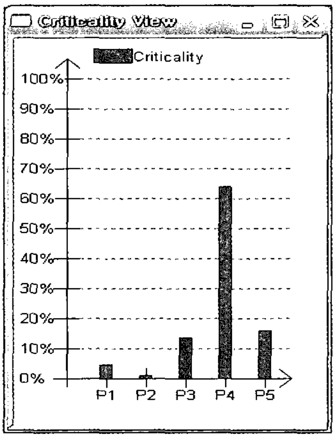
(Figure 14) shows the real time UPS workflow modeling, which consists of 5 workflow scenarios (P1, P2, P3, P4, and P5) of UPS system such as Normal return (P1), Overload (P2), Service interruption(P3), Normal status (P4), and Failure (P5) scenario. As a result, we can recognize the critical scenario (path) 'P4', that is, the normal status, having the critical value (0.64).

This tool can also show the bar graph to display the criticality of all possible component scenarios in (figure 15). There displays the bar graphs from high one to low criticality, such as P4 > P5 > P3 > P1 > P2.



No	PATH	LENGTH	CRITICALITY
P1	START -> Use Battery -> Input Filter -> Input Transform -> Rectify -> Invert -> Output Transform -> Synchronize -> Output Filter -> E...	8	0.040000
P2	START -> Use Battery -> Input Filter -> Input Transform -> Synchronize -> Output Filter -> END	5	0.012000
P3	START -> Use Battery -> Invert -> Output Transform -> Synchronize -> Output Filter -> END	5	0.140000
P4	START -> Input Filter -> Input Transform -> Rectify -> Invert -> Output Transform -> Synchronize -> Output Filter -> END	7	0.640000
P5	START -> Input Filter -> Input Transform -> Synchronize -> Output Filter -> END	4	0.160000

(Fig. 14) Components associated within 5 sub-workflows of UPS



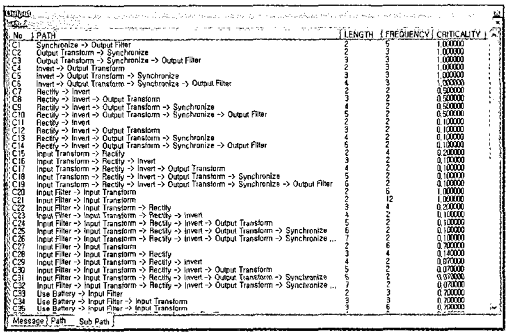
(Fig. 15) Criticality of all component scenarios

In order easily to recognize common component and cluster of components, it is arranged each sub workflow and identify diverse components in figure 16. These diverse components may be reused when developing new UPS system. As a result, it may be very easily and fast to develop new quality one.

(Figure 16) show only to appear diverse components (such as component, process component, diverse clusters of components) through the automatic conversion from WODA tool.

Figure 16 shows the result of subpaths.

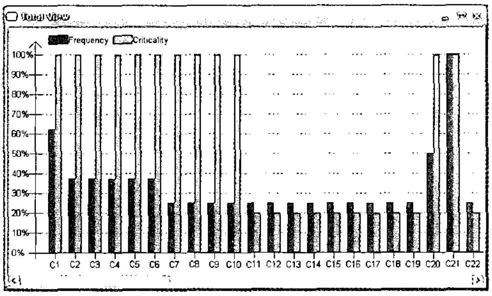
Subpath is the reusable group of components as a subset of the possible component scenarios. In Sub-Path, we add one more function for checking frequency, that is, how many times the component/the group of components use. There are extracted 69 components of the whole UPS component scenarios.



No	PATH	LENGTH	FREQUENCY	CRITICALITY
C1	Synchronize -> Output Filter	2	5	1,000000
C2	Output Transform -> Synchronize	2	3	1,000000
C3	Output Transform -> Synchronize -> Output Filter	3	3	1,000000
C4	Invert -> Output Transform	2	3	1,000000
C5	Invert -> Output Transform -> Synchronize	3	3	1,000000
C6	Invert -> Output Transform -> Synchronize -> Output Filter	4	3	1,000000
C7	Rectify -> Invert	2	2	0,500000
C8	Rectify -> Invert -> Output Transform	3	2	0,500000
C9	Rectify -> Invert -> Output Transform -> Synchronize	4	2	0,500000
C10	Rectify -> Invert -> Output Transform -> Synchronize -> Output Filter	5	2	0,500000
C11	Rectify -> Invert	2	2	0,100000
C12	Rectify -> Invert -> Output Transform	3	2	0,100000
C13	Rectify -> Invert -> Output Transform -> Synchronize	4	2	0,100000
C14	Rectify -> Invert -> Output Transform -> Synchronize -> Output Filter	5	2	0,100000
C15	Input Transform -> Rectify	2	4	0,200000
C16	Input Transform -> Rectify -> Invert	3	2	0,100000
C17	Input Transform -> Rectify -> Invert -> Output Transform	4	2	0,100000
C18	Input Transform -> Rectify -> Invert -> Output Transform -> Synchronize	5	2	0,100000
C19	Input Transform -> Rectify -> Invert -> Output Transform -> Synchronize -> Output Filter	6	2	0,100000
C20	Input Filter -> Input Transform	2	6	1,000000
C21	Input Filter -> Input Transform	2	12	1,000000
C22	Input Filter -> Input Transform -> Rectify	3	4	0,200000
C23	Input Filter -> Input Transform -> Rectify -> Invert	4	2	0,100000
C24	Input Filter -> Input Transform -> Rectify -> Invert -> Output Transform	5	2	0,100000
C25	Input Filter -> Input Transform -> Rectify -> Invert -> Output Transform -> Synchronize	6	2	0,100000
C26	Input Filter -> Input Transform -> Rectify -> Invert -> Output Transform -> Synchronize ...	7	2	0,100000
C27	Input Filter -> Input Transform	2	6	0,700000
C28	Input Filter -> Input Transform -> Rectify	3	4	0,140000
C29	Input Filter -> Input Transform -> Rectify -> Invert	4	2	0,070000
C30	Input Filter -> Input Transform -> Rectify -> Invert -> Output Transform	5	2	0,070000
C31	Input Filter -> Input Transform -> Rectify -> Invert -> Output Transform -> Synchronize	6	2	0,070000
C32	Input Filter -> Input Transform -> Rectify -> Invert -> Output Transform -> Synchronize ...	7	2	0,070000
C33	Use Battery -> Input Filter	2	3	0,700000
C34	Use Battery -> Input Filter -> Input Transform	3	3	0,700000
C35	Use Battery -> Input Filter -> Input Transform	3	6	0,700000

(Fig. 16) All reusable components/groups

(Figure 17) displays the result of the figure 16 with the bar graphs. In this figure 16, the blue color is the value of Frequency, and the red color is the value of Criticality.



(Fig. 17) Graph of all reusable components/groups

There are various clustered (grouped) the components for reusability.

6. Conclusion

Through the domain based analysis we identify common/uncommon process components (or cluster of components) focusing more on dynamic modeling aspects of systems. we also recognize the reusable numbers of the particular components with the criticality or frequency of common/uncommon components (or cluster of them). We suggest component test plan metrics for reusability measurement to enhances productivity for new right systems. The proposed reusability metrics are employed to improve the productivity of component test plan metrics through component scenario prioritization.

Now, We are developing our case tool to work component development for a new system easier, faster, and more stable, while determining the importance of measure weighted components.

References

[1] http://www.dtc.army.mil/tts/2003/proceed/hudgins/

[2] George T. Heineman, William T. Councill, Component Based Software Engineering, Addison Wesley 2001.

[3] Musa, J.D. The Operation Profile in Software Reliability Engineering: An Overview, AT&T Bell Laboratories Murray Hill, NJ, pp.140-154, 1992.

[4] Musa, J.D. Operational Profiles in Software Reliability Engineering, AT&T Bell

Laboratories, pp.14-32, 1993.

[5] Prieto Diaz, “Domain Analysis: An Introduction,” Software Engineering Notes 15, 1990.

[6] Y. Kim, E. Choi, B. Jeon, “A Study on Component Architecture and Platform Technology for Defense Software,” Agency for Defense Development, 2003.

[7] Griss, Domain engineering and Reuse, IEEE, 1999

[8] Prieto Diaz, Domain Analysis: An Introduction. Software Engineering Notes 15, 1990

[9] Mealy, G.H. A Method for Synthesizing Sequential Circuits. Bell System Technical Journal, Vol. 34, pp. 1045-1079, 1956.

[10] Moore, E.F. Gedanken Experiments on Sequential Machines. In Automata Studies. Annals of Mathematical Studies #34. NJ, 1956.

[11] Beizer, B. Black-Box Testing: Techniques for Functional Testing of Software and System, John Wiley&Sons, NY 1995.

[12] Patrick Donohoe, Software Product Lines, Kluwer Academic Publishers, 2000, pp 3-22.

[13] Kim, YoungChul, Carlson, C.R "Adative Design Based Testing for OO Software", ISCA 15th International Conference on Computers and Their Applications (CATA-2000), New Orleans, Louisiana, March 2000.

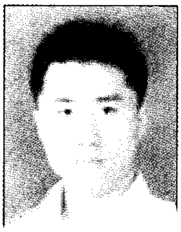
[14] IEEE Standard 982.2-1987 Guide for the Use of Standard Dictionary of Measures to Produce Reliable Software.

[15] Linda Rosenberg, Ted Hammer, Jack Shaw, "Software Metrics and Reliability", 9th International Symposium on Software Reliability Engineering, Germany, Nov. 1998.

The Authors

김 영 철

2000년 Illinois Institute of Technology(공학박사)
2000년~2001년 LG 산전 중앙연구소 Embedded system 부장
2001년- 현재 홍익대학교 컴퓨터정보통신 조교수
관심분야 : 테스트 성숙도 모델(TMM), Use Case 방법론 및 도구 개발, BPM, 사용자 행위 분석 방법론, Embedded SE
이 메 일 : bob@selab.hongik.ac.kr



김 우 열

2004년 홍익대학교 컴퓨터정보통신(학사)
2006년 홍익대학교 일반대학원 소프트웨어공학전공(석사)
2006년- 현재 홍익대학교 일반대학원 박사과정
관심분야 : 상호운용성, 임베디드 소프트웨어 개발 방법론 및 도구 개발, 컴포넌트 시험 및 평가, 리팩토링
이 메 일 : john@selab.hongik.ac.kr

서 윤 숙

1999년 홍익대학교 전자전산학과(학사)
2005년 홍익대학교 일반대학원 소프트웨어공학전공 졸업(석사)
2005년 홍익대학교 일반대학원 소프트웨어공학전공 박사과정
관심분야 : 사용자 행위 분석 방법론, 비즈니스 프로세스 모델링(BPM), 컴포넌트 기반 개발
이 메 일 : jyun@selab.hongik.ac.kr

김 기 두

2003년 홍익대학교 컴퓨터정보통신과(학사)
2005년 홍익대학교 일반대학원 소프트웨어공학전공 졸업(석사)
2005년 홍익대학교 일반대학원 소프트웨어공학전공 박사과정
2005년- 현재 한국정보통신기술협회 SW시험인증연구소 연구원
관심분야 : 테스트 성숙도 모델, 테스트 프로세스, 임베디드 소프트웨어 테스트, 소프트웨어 신뢰성 테스트
이 메 일 : kdkim@tta.or.kr