Journal of KIIT. Vol. 15, No. 11, pp. 103-109, Nov. 30, 2017. pISSN 1598-8619, eISSN 2093-7571 103
http://dx.doi.org/10.14801/jkiit.2017.15.11.103

소프트웨어 품질 속성의 인식을 위한 적응성 네트워크 모델

김 은 미*

Adaptive Network Model for the Recognition of Software Quality Attributes

Eun-Mi Kim*

호원대학교 교내학술연구비의 지원을 받았음.

요 약

ICT와 비즈니스 환경의 변화에 원활하게 대응할 수 있는 SOA와 같은 소프트웨어 구축 방법론이 활용되고 있는 상황에서, 기존의 정형화된 소프트웨어 품질 속성이 아니라 지속적인 환경변화에 능동적으로 적응할 수 있는 소프트웨어 품질 모델 또는 품질 속성의 운용체계가 요구된다. 인터넷 상의 웹과 같은 네트워크를 바탕으로 SOA가 구현되면 모든 이해관계자들이 실시간으로 상호작용 할 수 있으므로 환경의 변화에 따른 다양한 요구와 반응들을 통합하고 범주화하여 새로운 품질 요소들을 획득할 수 있다. 본 논문에서는 소프트웨어 품질 요소들을 계통적으로 구조화한 'S.Q.C. 범주'(소프트웨어 품질 특성 범주)를 정의하고 각 이해관계자들이 네트워크를 통하여 그 구성 원소들을 공유하고 상호 참조함으로써 자신들의 품질 필요성에 따라 지속적으로 소프트웨어 품질 속성을 갱신하고 확장할 수 있는 집단적 적응성 범주화 모델을 제시한다.

Abstract

With the use of software-building methodologies such as SOA, which can respond seamlessly to the changes in ICT and business environment, there is a need for a software quality model or quality attribute operating system that can actively adapt to continuous environmental changes, rather than existing stereotyped software quality attributes. When an SOA system is implemented on a network such as the web on the Internet, all stakeholders can interact with each other in real time, so that new quality factors can be acquired by integrating and categorizing various demands and responses according to changes in the environment. In this paper, we propose a collective model of adaptive categorization, where 'S.Q.C. category'(category of software quality characteristics), a systematic structuring of software quality factors, is defined and each stakeholder shares and cross-references its constituent elements over the network to renew and expand the software quality attributes continually.

Keywords
software quality attributes, SOA, stakeholders, collective model, ddaptive categorization

* 호원대학교 컴퓨터·게임학과
- ORCID: http://orcid.org/0000-0003-2108-8809

· Received: Aug. 28, 2017, Revised: Oct. 18, 2017, Accepted: Oct. 21, 2017
· Corresponding Author: Eun-Mi Kim
Dept. of Computer & Game, Howon University, 64 Howondae 3gil, Impi, Gunsan, Jeollabuk-Do, 54058, Korea,
Tel.: +82-63-450-7543, Email: ekim@howon.ac.kr

Ⅰ. 서 론

ASP(Application Service Provider) 등에 의한 소프트웨어 아웃소싱은 인터넷과 웹 관련 기술의 발전으로 그 활용 가능성과 범위가 크게 확대되고 있다. 특히 XML에 기반으로 SOAP(Simple Object Access Protocol), WSDL(Web Services Description Language), UDDI(Universal Description, Discovery and Integration) 등의 프로토콜 스택을 구성하여 서비스 지향적 분산 컴퓨팅 기술을 구현하는 웹 서비스 기술이 발전하였고, 이를 통하여 SOA(Service Oriented Architecture)의 개념이 접목됨으로써 소위 SaaS(Software as a Service)라는 서비스 기반의 소프트웨어 운용체계로도 확장되고 있다[1].

특정 대상에 대한 아웃소싱 서비스에서 비롯된 ASP를 불특정 다수를 대상으로 하는 서비스로까지 확장한 SaaS는 다양한 사용자의 요구에 대한 맞춤형 서비스로의 확장성을 특징으로 한다. 이는 공유와 재사용이 가능한 소프트웨어를 서비스 단위 또는 컴포넌트로서 연동하고 통합하여 누구나 이용 가능한 서비스를 구성할 수 있는 아키텍처인 SOA 또는 웹 서비스의 특성을 반영한다할 것이다. SOA의 개념을 기본 전제로 서비스 공급자는 다른 서비스의 사용자일 수 있고, 서비스 사용자는 새로운 서비스의 공급자가 될 수 있다. 따라서 소프트웨어 서비스 사용자의 역할과 비중이 더욱 강화되는 사용자 중심의 개방적 환경에서 이러한 소프트웨어 개발 및 운용 환경의 변화는 사용자의 참여가 SOA의 결정적 요소가 될 수 있음을 보여준다[2].

SOA를 기반으로 하는 소프트웨어들의 품질과 관련된 활동에서도 서비스 사용자를 포함한 모든 이해관계자들(Stakeholders)이 적절하고 적극적으로 참여할 수 있는 체계를 마련할 수 있다면 이것은 매우 중요한 서비스 운용 수단의 한 부분이 될 것이다. SOA 시스템에서 새로운 서비스를 구축하는 프로젝트를 수행하게 된다면 일반적으로 이 '프로젝트 서비스'를 구성하기 위하여 서비스 공급자들이 서비스 레지스트리(또는 브로커)에 등록한 '컴포넌트 서비스'들을 사용하게 될 것이다. 이러한 '컴포넌트 서비스'들에 대한 요구나 필요성은 '프로젝트 서비스'에 대한 비즈니스상의 목적과 필요성에 의하여 결정될 것이다. 즉 SOA 시스템에서 서비스를 구성하는 소프트웨어의 품질에 대한 요구사항(Software Quality Requirements)이나 필요성은 사용자 또는 이해관계자의 비즈니스적 관점에 따라 달라진다. 따라서 서비스 제공자가 사용자의 비즈니스상의 목적과 필요성에 무관하게 독립적으로 자신이 제공하는 서비스에 대한 소프트웨어 품질 모델을 특정하여 서비스 레지스트리에 등록하더라도, 이런 서비스들을 '컴포넌트 서비스'로 하여 구성되는 '프로젝트 서비스'의 소프트웨어 품질 모델은 각 '컴포넌트 서비스'에 대하여 등록된 위의 품질 모델들을 단순히 통합함으로써 구성되지 않을 수 있다. 비즈니스적 관점에서 각 '컴포넌트 서비스'의 소프트웨어 품질 속성이 모두 필요하지는 않을 수 있고, 각 '컴포넌트 서비스'를 네트워크상에서 연동하는 SOA의 아키텍처적 특성에 따라 요구되는 품질 속성들이 비즈니스 상의 필요성에 맞추어 추가될 수도 있다[3].

SOA 시스템에서와 같이 서비스를 구성하는 각 소프트웨어들에 대한 정보가 등록되고 공유되어 활용될 수 있다면, 서비스 사용자뿐만 아니라 모든 유형의 이해관계자들이 자신들의 비즈니스적 관점에 따라 특정 서비스에 대하여 개인화된 맞춤형의 소프트웨어 품질 모델을 구축할 수 있을 것이다. 이러한 개인화된 품질 모델들이 공유되고 통합적으로 운용된다면 가장 타당한 소프트웨어 품질 모델과 품질 속성을 이끌어 내고 선택할 수 있다. 특히 기술적 조건이나 비즈니스적 필요성 등의 환경이 변화하더라도 이에 상응하여 각 이해관계자들에 의하여 새로운 품질 모델이 제시되고, 공유되어 통합된다면 이러한 환경의 변화가 반영된 새로운 품질 모델이나 품질 속성을 끌어낼 수 있다.

본 논문에서는 SOA의 서비스에 대하여 특정한 소프트웨어 품질 모델이나 품질 속성이 필요하다거나 적합함을 보여주는[4][5] 대신에 모든 이해관계자들이 자신의 비즈니스적 필요성에 따라 소프트웨어 품질 모델들을 구성하여 이들을 공유하고 통합할 수 있는 방법론과 프로세스를 제시할 것이다. 이는 이해관계자들의 네트워크를 통하여 사용 환경의 변화에 적합한 품질 속성을 추출하고 지속적으로

수정, 보완하는 적응성 있는 네트워크 모델이다.

소프트웨어 품질 모델에 대해서는, 품질 속성만으로 이를 구성하는 대신에 각 품질 속성에 대응하는 품질 관련 요소들인 품질 척도, 품질 요구사항, 품질 필요성 등을 계통적 패키지로 구조화한 S.Q.C. 범주(Software Quality Characteric Category; 소프트웨어 품질 특성 범주)를 정의하여 사용한다. 이 S.Q.C. 범주는 소프트웨어 품질관련 요소들을 계통적 정보로 제시한다. 이는 각 요소들마다 별개로 분류하는 것보다 더 효과적인 범주화와 분류가 가능하고 이를 통하여 소프트웨어 품질 모델을 포함하고 있는 S.Q.C. 범주 모델을 구성한다. 이를 위해서 각 이해관계자들이 생성하거나 보완한 S.Q.C. 범주들은 S.Q.E.DB(Software Quality Evaluation DB; 소프트웨어 품질평가 데이터베이스)와 같은 웹 또는 네트워크상에서 운영되는 데이터베이스에서 공유되고 통합된다.

다음 장에서는 S.Q.C. 범주의 수학적 구조를 설명한다. 또한, 이 S.Q.C. 범주들을 비즈니스 상의 필요성에 맞게 이해관계자들이 참여하여 직접 수정, 보완, 생성하는 절차인 Q.C.Rec.(S.Q.C. Recategorization; 소프트웨어 품질 특성 재범주화) 프로세스의 실행에 필요한 구성 요소와 조건들에 대하여 살펴본다. 3장에서는 이 Q.C.Rec. 프로세스의 구체적인 단계와 절차에 대하여 기술한다. 마지막 장에서는 본 연구 결과의 의미와 그에 따른 향후의 연구방향에 대하여 논한다.

II. S.Q.C. 범주의 구조 및 구현 시스템

2.1 S.Q.C. 범주의 수학적 구조

하나의 S.Q.C. 범주 $\mathcal{C}$는 네 개의 동치(Equivalence) 클래스 N, R, M 그리고 A를 그 대상으로 갖는 범주이다[6]. 즉 $\mathcal{C}$의 대상들의 클래스인 $\mathbf{Ob}(\mathcal{C})$는

$$\mathbf{Ob}(\mathcal{C}) = \{N,\ R,\ M,\ A\} \tag{1}$$

인데, 여기서 N, R, M 그리고 A는 각각 소프트웨어 품질 필요성, 품질 요구사항, 품질 척도, 품질 속성들의 동치 클래스들이다. 이 대상들 사이의 모든 사상(Morphism)들의 클래스인 $\mathbf{Hom}(\mathcal{C})$는

$$\mathbf{Hom}(\mathcal{C}) = \cup_{X,\,Y\in \mathbf{Ob}(\mathcal{C})} \mathrm{Hom}(X,\,Y) \tag{2}$$

으로 표현되는데, $\mathrm{Hom}(X,\,Y)$는 X에서 Y로 가는 모든 사상들의 클래스이다. $f\in \mathrm{Hom}(X,\,Y)$ 인 임의의 사상 f가 $x\in X$ 인 x를 $y\in Y$ 인 y에 대응시킨다면, 즉 $f: x\to y$ 는 어떤 이해관계자가 품질요소 x를 y에 대응시키는 하나의 사례를 나타낸다.

각 동치 클래스 N, R, M 그리고 A를 정의 하는 동치 관계 '$\sim_{\mathcal{C}}$'는 다음과 같이 정의된다. $X,\,Y\in \mathbf{Ob}(\mathcal{C})$ 이고 $x_1, x_2\in X$ 이며 $y\in Y$인 모든 y에 대하여

$$x_1 \sim_{\mathcal{C}} x_2 \Leftrightarrow \\ \exists f, g\in \mathbf{Hom}(\mathcal{C})\ f: x_1\to y,\ g: x_2\to y. \tag{3}$$

또 다른 S.Q.C. 범주들인 $\mathcal{C}'$, $\mathcal{C}''$ … 등이 일반적으로 존재하고, 따라서 동일한 유형(품질 요소)의 대상들인 X', X'' … 등의 동치 클래스들도 존재하므로 확장된 클래스인 $\tilde{X}= X\cup X'\cup X''\cup\cdots$ 가 정의된다. 각 S.Q.C. 범주들에서 각 X, Y와 동일한 유형의 동치 클래스들 사이의 사상들인 f, f', f'' …들을 조합하면 확장된 클래스들인 $\tilde{X}$, $\tilde{Y}$ 사이의 사상인 $\tilde{f}: \tilde{X}\to \tilde{Y}$ 가 정의된다. 이렇게 정의된 사상들의 클래스는 $\tilde{X}$, $\tilde{Y}$ 등을 대상으로 갖는 '소프트웨어 품질 특성 범주' $\tilde{\mathcal{C}}$의 사상들의 클래스인 $\mathbf{Hom}(\tilde{\mathcal{C}})$를 정의한다. 즉 $\tilde{f}\in \mathbf{Hom}(\tilde{\mathcal{C}})$ 이고,

$$\mathbf{Ob}(\tilde{\mathcal{C}}) = \{\tilde{N},\ \tilde{R},\ \tilde{M},\ \tilde{A}\}. \tag{4}$$

식 (3)의 사상 f 등에 의한 동치 관계 '$\sim_{\mathcal{C}}$'는 $\tilde{f}$ 등에 의한 동치 관계 '$\sim_{\tilde{\mathcal{C}}}$'로 대체되고, 이 동치 관계에 의하여 각 품질 요소의 클래스 $\tilde{X}$는 각 S.Q.C. 범주들의 동치 클래스인 X, X', X'' …들로 분할된다.

각 이해관계자들이 참여하여 수행하는 역할은 비즈니스 상의 필요성에 맞추어 각 S.Q.C. 범주들에 속하는 S.Q.C. 4-터플(4-tuple),

$$q_C=(n,\ r,\ m,\ a) \tag{5}$$

들을 해당 범주의 각 동치 클래스들의 원소인 $n\in N$, $r\in R$, $m\in M$, $a\in A$ 들로 구성하는 것이다. 이 S.Q.C. 4-터플은 어떤 이해관계자가 소프트웨어 품질 필요성 n에 따른 품질 요구사항 r을 품질 척도 m에 의한 품질 속성 a로 대응하는 사례에 해당한다. 이 4-터플의 각 성분은 해당 동치 클래스들을 생성하거나 보완하게 된다. 일반적으로 품질 속성 성분 a는 단순한 품질 속성 대신에 계층구조상 그 상위 개념들인 소프트웨어 품질 특성들의 계통을 포함하는 S.Q.C. 속성을 사용한다(그림 1).

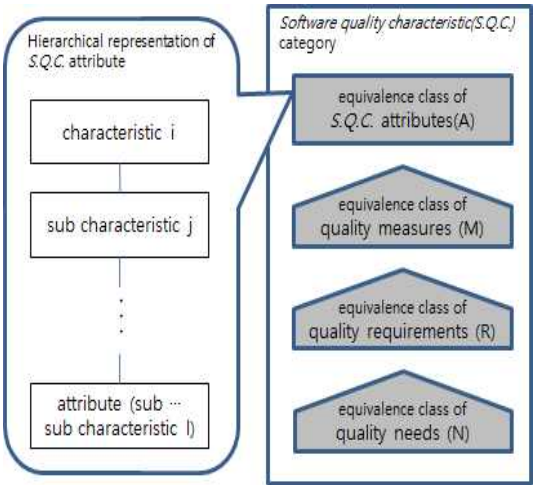

그림 1. S.Q.C 범주와 S.Q.C 속성의 계층구조
Fig. 1. S.Q.C category and the hierarchical representation of S.Q.C attributes

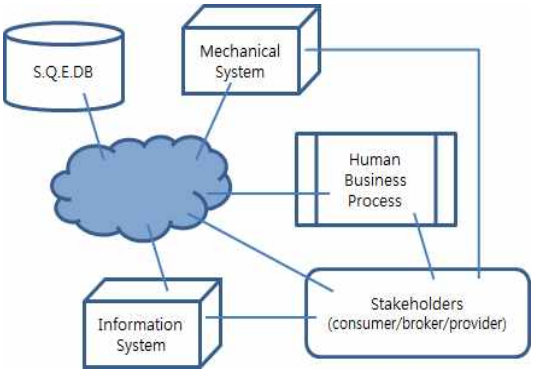

그림 2. Q.C.Rec. 프로세스의 네트워크 구성요소
Fig. 2. Network components of Q.C.Rec. process

이해관계자들에 의한 S.Q.C. 4-터플의 구성은 $\mathrm{Hom}(\tilde{C})$의 사상들에 의하여 기술되는데, 일반적으로 이 사상들에 따라서 위의 동치관계 '$\sim_{\tilde{C}}$'에 의한 $\mathrm{Ob}(\tilde{C})$의 분할이 달라진다. 서로 다른 분할은 각각 서로 다른 S.Q.C. 범주 모델에 해당하고, 이는 서로 다른 소프트웨어 품질 모델로 나타난다. 따라서 각 이해관계자가 $\mathrm{Ob}(\tilde{C})$의 특정한 분할을 유지하며 S.Q.C. 4-터플들을 구성하는 프로세스는 일관된 하나의 품질모델을 구축하는 것임이 논리적으로 입증된다.

2.2 Q.C.Rec. 프로세스의 실행 요건

SOA 시스템에서 이해관계자들이 파악한 소프트웨어 품질 필요성에 따라 그 품질을 평가하기 위하여 각 이해관계자들이 참여하여 생성하는 품질 관련데이터들을 공유하고 활용할 수 있도록 S.Q.E.DB와 같은 데이터베이스를 웹 또는 네트워크상에서 운영한다[7].

지속적으로 변화하는 기술과 비즈니스 환경의 요청에 맞추어 새로운 소프트웨어 품질 모델이나 품질 속성을 구성할 수 있는 Q.C.Rec. 프로세스를 실행하는 과정에서, 각 이해관계자들이 S.Q.C. 4-터플을 생성하기 위하여 참조하는 S.Q.C. 범주들의 포괄적인 기준 모델인 C.R.Q.C.(Comprehensive Reference model of S.Q.C Categories)를 위의 S.Q.E.DB와 같은 데이터베이스에 등록하여 운용한다. 또한 서비스 레지스트리에 등록된 각 소프트웨어들은 새로운 서비스를 구축하는 프로젝트에서 컴포넌트로 활용될 것이므로 이들의 S.Q.C. 범주 모델인 Q.C.S.C.(S.Q.C. Category Model of Service Component Software)들이 위 C.R.Q.C.를 바탕으로 구성되어 S.Q.E.DB에 등록될 수 있다.

이 C.R.Q.C.와 Q.C.S.C.들은 최초의 단계에서는 ISO/IEC 25000 SQuaRE 시리즈와 같은 잘 알려진 표준을 바탕으로 S.Q.C. 범주들을 구성함으로써 작성될 수 있고, 새로운 서비스를 구축하는 과정에서는 이미 수행된 Q.C.Rec. 프로세스에 의하여 갱신된 상태이다. 이러한 상태를 초기 조건으로 하여 새로운 Q.C.Rec. 프로세스들이 새로운 소프트웨어 서비스를 구축, 운용하는 과정에서 실행된다.

III. Q.C.Rec. 프로세스

Q.C.Rec. 프로세스는 새로운 소프트웨어 서비스를 구축하는 프로젝트의 비즈니스적 요청에 따라 각 이해당사자가 S.Q.C. 4-터플을 생성하여 이를 범주화하고 분류함으로써 S.Q.C. 범주들을 수정, 보완, 생성한다. Q.C.Rec. 프로세스는 (그림 3)과 같이 3개의 페이즈(Phase)로 구성된다.

첫 번째 페이즈에서는 새로운 프로젝트 서비스에 대하여 이해관계자가 소프트웨어 품질 필요성을 문서화하고, 프로젝트 서비스 구성에 사용될 컴포넌트 서비스들을 확정한다. 이 후 S.Q.E.DB에 등록된 이들의 Q.C.S.C.들을 모아서 프로젝트 서비스의 '집합 Q.C.S.C'.(Assembled Q.C.S.C.)를 작성한다.

두 번째 페이즈에서는 프로젝트 서비스에 대한 품질 필요성에 해당하는 '집합 Q.C.S.C'.의 S.Q.C. 범주들을 수정, 보완함으로써 '집합 Q.C.S.C'를 갱신하고, 이에 대응되지 않고 남겨진 품질 필요성을 구분한다.

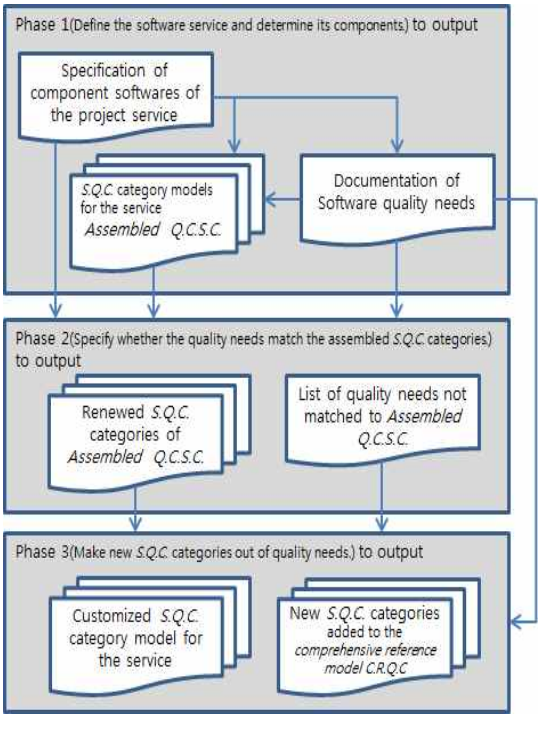

그림 3. Q.C.Rec. 프로세스의 세 페이즈와 그 출력들
Fig. 3. Three phases of Q.C.Rec. process and their outputs

세 번째 페이즈에서는 '집합 Q.C.S.C'에 대응되지 않고 남겨진 품질 필요성에 대응되는 C.R.Q.C.의 S.Q.C. 범주들을 갱신하고, 대응되지 않고 여전히 남는 품질 필요성에 대해서는 새로운 S.Q.C. 범주들을 생성하여 C.R.Q.C.에 추가함으로써 이를 갱신한다. 갱신되었거나 새로 생성된 S.Q.C. 범주들을 모아 프로젝트 서비스에 대한 맞춤형 S.Q.C. 범주 모델을 구성하고 프로젝트 서비스에 대한 Q.C.S.C.로 등록한다.

3.1 페이즈 1 : 소프트웨어 서비스를 정의하고 그 컴포넌트 서비스들을 결정

비지니스적 배경이나 기술적 조건 등을 바탕으로 SOA를 통해 소프트웨어 서비스를 구축하려는 프로젝트의 범위나 목표가 확고하게 결정되어야 한다.

Step 1.1 : 이용 가능한 소프트웨어들로 서비스를 구성할 수 있을 지 그 타당성을 조사한다. 비즈니스상의 목표와 필요성에서 소프트웨어 서비스로 구현할 수 있는 영역을 결정한다.

Step 1.2 : 컴포넌트 소프트웨어들의 목록과 SOA상의 그 제약들을 상술한다. 서비스 레지스트리에 등록된 소프트웨어들을 컴포넌트로 선정하여 새로운 서비스를 구성하고 소프트웨어 품질 필요성의 타당한 범위를 확정하기 위하여 소프트웨어 활용상의 제약들을 제시한다.

Step 1.3 : 소프트웨어 품질 필요성을 문서화한다. 구체적인 필요성이 드러날 수 있는 형태로 모든 소프트웨어 품질 필요성을 목록으로 작성한다.

Step 1.4 : 각 컴포넌트 소프트웨어들에 대한 Q.C.S.C.들을 모은다. S.Q.E.DB에 등록된 Q.C.S.C.들을 조합하여 프로젝트 서비스에 대한 '집합 Q.C.S.C'를 구성한다.

3.2 페이즈 2 : '집합 Q.C.S.C'와의 대응여부에 따라 소프트웨어 품질 필요성 구분 처리

페이즈 1에서 산출된 '집합 Q.C.S.C'과 소프트웨어 품질 필요성 목록을 비교하여 품질 필요성에 대응되는 S.Q.C. 범주들을 갱신하고, 대응되지 않고

남겨지는 품질 필요성들은 구분하여 정리한다.

Step 2.1 : 소프트웨어 품질 필요성에 부합하도록 '집합 Q.C.S.C'의 S.Q.C. 범주를 갱신한다.

[Step 2.1.1] S.Q.C. 속성을 수정, 보완한다.

[Step 2.1.2] 소프트웨어 품질척도와 그 기준을 수정, 보완한다.

[Step 2.1.3] 타당한 소프트웨어 품질 요구사항을 상술하고 품질 필요성을 보완한다.

Step 2.2 : '집합 Q.C.S.C'에 속하지 않은 소프트웨어 품질 필요성을 확인한다. '집합 Q.C.S.C'의 S.Q.C. 범주에 대응되지 않은 이 품질 필요성들의 목록을 따로 정리한다.

3.3 페이즈 3 : 소프트웨어 품질 필요성으로부터 새로운 S.Q.C. 범주 구성

페이즈 2에서 S.Q.C. 범주에 대응되지 않고 남은 소프트웨어 품질 필요성 중 C.R.Q.C.의 S.Q.C. 범주에 대응되는 것들에 대해서는 대응되는 S.Q.C. 범주들을 갱신하고 여전히 대응되지 않는 것들에 대해서는 새로운 S.Q.C. 범주들을 구성한다.

Step 3.1 : 등록된 S.Q.C. 범주와의 대응 여부에 따라 소프트웨어 품질 필요성을 구분하여 처리한다.

[Step 3.1.1] 소프트웨어 품질 필요성에 부합하도록 C.R.Q.C.의 S.Q.C. 범주를 갱신한다.

[[Step 3.1.1.1-3.1.1.3]] 소프트웨어 품질 필요성에 대응되는 C.R.Q.C.의 S.Q.C. 범주에 대하여 [Step 2.1.1-2.2.3]의 과정을 수행한다.

[Step 3.1.2] 등록된 S.Q.C. 범주에 대응되지 않는 소프트웨어 품질 필요성을 확인한다. C.R.Q.C.에 속하지 않은 이 필요성들의 목록을 따로 정리한다.

Step 3.2 : C.R.Q.C.에 대응되지 않은 소프트웨어 품질 필요성들로부터 새로운 S.Q.C. 범주들을 구성한다.

[Step 3.2.1] 소프트웨어 품질 필요성을 분류하고 그에 대응하는 품질 요구사항을 작성한다.

[Step 3.2.2] 소프트웨어 품질 척도와 그 기준을 작성한다.

[Step 3.2.3] S.Q.C. 속성을 구성한다.

Step 3.3 : 새로운 S.Q.C. 범주 모델들을 생성한다. 등록된 S.Q.C. 범주에 대응되지 않는 소프트웨어 품질 필요성으로부터 생성된 S.Q.C. 범주들을 추가하여 C.R.Q.C.를 갱신하고, 갱신되었거나 새로 생성된 S.Q.C. 범주들이 모여 프로젝트 서비스에 대한 S.Q.C. 범주 모델을 구성한다.

Ⅳ. 결 론

웹 기술의 발달로 다양한 이해관계자와의 즉각적인 상호작용이 가능해지고, 소프트웨어 품질과 관련해서도 이해관계자들의 다양한 반응들이 실시간으로 반영되거나 관련 활동에 직접 참여할 수 있는 기술적 환경이 조성되었다. 이러한 배경에서 본 논문은 소프트웨어 품질 요소들에 대하여 이해관계자들이 직접 참여하는 집단적 인식 시스템을 제안한 것이다. 소프트웨어 품질 전문가가 아닌 일반적인 이해관계자들도 효과적으로 품질요소들을 식별할 수 있도록 각 품질 요소들에 대한 데이터를 S.Q.C. 4-터플과 그 동치 클래스인 S.Q.C. 범주라는 구조화된 패키지로 제시하였다.

이해관계자들이 자신들이 파악한 소프트웨어 품질 필요성을 파악하고, 이미 등록된 품질 요소들에 대한 데이터들과 비교함으로써 기존의 품질 모델로 설명할 수 있는 것인지 아니면 새로운 품질 요소들을 구성해야 하는 것이지를 식별한다. 이러한 시스템이 성공적으로 작동하기 위해서는 이해관계자들과의 인터페이스를 효율적으로 구성하는 섬세한 기술적 요소들이 더불어 고려되어야 함은 당연하다. 특히 각 이해관계자가 Q.C.Rec. 프로세스를 정확히 수행함으로써 이를 통해 생성된 S.Q.C. 4-터플들로 구성되는 S.Q.C. 범주들이 $Ob(\tilde{C})$의 동일한 분할 구조(S.Q.C. 범주 모델)에 속하도록 제한하는 기능과 함께, 서로 다른 이해관계자들이 생성한 S.Q.C. 범주 모델 가운데 하나를 선택할 수 있도록 진화알고리듬과 같은 기능을 적용할 수도 있을 것이다. 더욱 전향적으로는, 이해관계자들의 인터페이스를 고도화하기 위하여 지능형 인터페이스 기술[8]을 접목하는 것도 바람직한데, 더 나아가 제안된 시스템을 네트워크에 기반을 둔 집단적 지능형 시스템으로 발전시키는 방안들을 마련하는 것도 의미가 있을 것이다.

References

[1] Atif F. Mohammad, "SOA 3.0, Cloud Computing and SaaS: A New Frontier", CONF-IRM 2009 Proceedings. 53, 2009.

[2] S. W. Choi, J. S. Her, and S. D. Kim, "Modeling QoS Attributes and Metrics for Evaluating Services in SOA Considering Consumers' Perspective as the First Class Requirement", IEEE Asia-Pacific Services Computing Conference, Dec. 2007.

[3] P. Bianco, R. Kotermanski, and P. Merson, "Evaluating a Service-Oriented Architecture", Technical Report CMU/SEI-2007-TR-015 ESC-TR-2007-015, Sep. 2007.

[4] L. O'Brien, L, P. Merson, and L. Bass, "Quality Attributes for Service-Oriented Architectures", International Workshop on Systems Development in SOA Environments (SDSOA'07), Jun. 2007.

[5] E. Kim and Y. Lee, "Quality model for web services", OASIS, 2005.

[6] M. Barr and C. Wells "Category Theory for Computing Science", Reprints in Theory and Applications of Categories, No. 22, pp. 1-538, 2012.

[7] Eun-Mi Kim and Cherl-Soo Pahk, "Dynamic Framework of Quality Evaluation for Web-based SOA", Journal of KIIT, Vol. 11, No. 9, pp. 143-150, Sep. 2013.

[8] M. Hartmann, "Challenges in Developing User-Adaptive Intelligent User Interfaces", Proceedings of the 17th Workshop on Adaptively and User Modeling in Interactive System, Darmstadt, Germany, pp. 6-11, Jan. 2009.

저자소개

김 은 미 (Eun-Mi KIM)

1993년 8월 : 전북대학교 전산통계학과(이학석사)
1997년 3월 : 오사카대학교 정보공학과(공학박사)
1997년 9월 ~ 현재 : 호원대학교 컴퓨터·게임학과 교수
관심분야 : 소프트웨어 품질 평가, 소프트웨어개발 방법론, 지능적 정보처리 등