

Z를 이용한 컴포넌트 정형 명세 활동의 제안

장 종 표[†] · 이 상 준^{**} · 김 병 기^{***}

요 약

소프트웨어 위기를 해결하기 위한 한가지 대응책으로 최근 컴포넌트기반 소프트웨어공학(혹은 컴포넌트웨어)이 등장하였다. 컴포넌트기반 소프트웨어공학을 이루는 관련 기술들 중 하나로써 컴포넌트기반 소프트웨어 개발 프로세스들이 제안되고 있다. 컴포넌트기반 소프트웨어 개발 프로세스중 컴포넌트 명세활동은 재사용 가능한 컴포넌트를 생성하기 위해서 꼭 필요하며 매우 중요한 활동이다. 그러나, 기존의 컴포넌트기반 소프트웨어 개발 프로세스에서는 컴포넌트 명세를 자연어로 기술함으로써 명세에 있어서 정확성을 갖지 못했고 사용자의 요구사항이 제대로 반영되었는지 검증할 수 있는 방법을 제공하지 못하고 있다.

본 논문에서는 정형방법에서 사용하고 있는 분석력과 논리성이 검증된 정형 명세 언어 Z를 이용하여 컴포넌트를 명세 하는 정형 활동을 제안하였다. 제안 활동은 5개의 태스크와 12개의 서브태스크들로 구성되며 10개의 산출물을 구체적으로 제시함으로써 재사용을 위한 컴포넌트 기반 소프트웨어 개발의 기초가 되는 컴포넌트 정형 명세 활동을 제안하였다. 제안 활동은 컴포넌트기반 소프트웨어 개발 프로세스의 초기단계에 사용자의 요구사항 명세의 정확성과 검증을 통하여 컴포넌트의 품질을 향상시킬 수 있다.

키워드 : 컴포넌트기반 소프트웨어공학, 컴포넌트기반 개발 프로세스, 정형방법, Z

A Proposal for Component Formal Specification Activities Using Z

Jong-Pyo Jang[†] · Sang-Jun Lee^{**} · Byung-Ki Kim^{***}

ABSTRACT

As a plan to counteract solving software crisis, the technology that is called Component Based Software Engineering (CBSE or Componentware) is introduced. Component Based Software Development (CBSD) Processes as one of the techniques among CBSE have been proposed. Component specification activities among CBSD Processes are necessary and important for generating reusable components. However, the existing CBSD Processes can't ensure the correctness in specifications because component specifications are written in natural language and don't provide a method which verifies that user requirements are reflected. In this paper, we propose that formal activities which specify component using formal specification language Z which is verified with the ability of analysis and logicity. The proposed activities are composed of 5 tasks and 12 subtasks. By presenting specific 10 products, it is also proposed that component specification activities which are the base of CBSD for reusing. The proposed activities are able to improve qualities of component through correctness and verification of requirements specification of users in the early step of component-based software process.

Key word : CBSE, CBD Process, Formal Methods, Z

1. 서 론

소프트웨어 생산성이 사용자들의 서비스에 대한 요구를 만족시키지 못했고, 소프트웨어 품질이 향상되지 않았으며, 소프트웨어의 개발과 유지보수 비용의 증가로 인해 소프트웨어 위기 문제가 발생하게 되었다. 이러한 위기를 해결하기 위한 한가지 대응책으로 최근 컴포넌트웨어(Componentware) 혹은 컴포넌트기반 소프트웨어 공학(CBSE, Component Based Software Engineering)이 등장하였다. CBSE를 이루는 관련 기술들 중 하나로써 컴포넌트기반 소프트웨어 개발 프로세스들이 제안되고 있다[1-3]. 일반적으로

소프트웨어 개발 프로세스는 사용자의 요구사항정의, 명세, 설계, 구현, 운용 및 유지보수 활동들로 이루어진다.

특히, 재사용을 목적으로 하는 컴포넌트기반 소프트웨어 개발 프로세스에서 컴포넌트의 품질은 매우 중요하다. 컴포넌트 개발 프로세스의 초기활동, 즉 명세활동에서 명세의 정확성과 명세의 검증을 통하여 에러를 찾아낸다는 것은 컴포넌트의 전체 품질에 매우 중요한 의미를 갖는다[4,5]. 그러나, 기존의 컴포넌트 기반 소프트웨어 개발 프로세스에서는 무엇이 전달되고 있는가에 대하여 명확성이 결여된 자연어를 이용하여 명세활동을 수행하였고, 명세활동에 있어서 품질관리는 검사(review)만을 통해서 이루어졌다. 이러한 문제의 해결책으로 명세 활동의 정형화가 필요하다.

본 논문에서는 컴포넌트기반 소프트웨어 개발 프로세스의 초기활동, 즉 명세활동에서 컴포넌트의 품질을 높일 수

† 준 회 원 : 전남대학교 전산학과 박사과정, (전남대) 정보통신연구소 연구원

** 정 회 원 : 서남대학교 컴퓨터영상정보통신학부 교수

*** 중신회원 : 전남대학교 컴퓨터정보학부 교수

논문접수 : 2000년 8월 28일, 심사완료 : 2000년 12월 29일

이도록 분석력과 논리성이 검증된 정형 명세 언어 Z를 이용한 컴포넌트 정형 명세 활동에 대하여 제안하였다. 논문의 구성은 다음과 같다. 2장에서는 기존의 컴포넌트기반 개발 프로세스에서의 명세 활동들을 살펴보고, 본 논문에서 사용하고 있는 정형 방법의 소개와 정형 방법에서 사용하고 있는 정형 명세 언어 Z에 대하여 정리하였다. 3장에서는 컴포넌트 명세를 정형화할 수 있는 구체적인 태스크들로 구성된 컴포넌트 정형 명세 활동을 제안하고 기존의 컴포넌트기반 소프트웨어 개발 프로세스의 명세활동과 비교하였고, 4장에서는 결론 및 향후연구방향에 대하여 논의하였다.

2. 관련연구

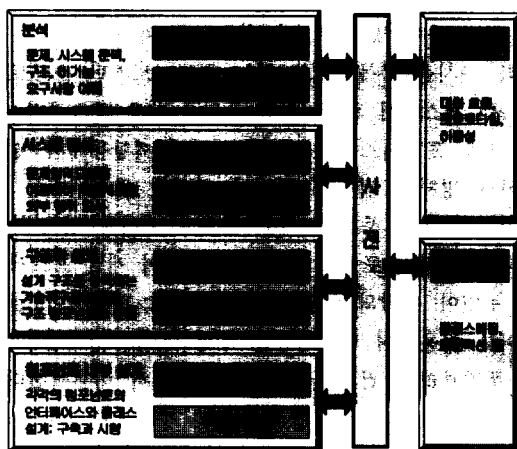
명세활동이란 소프트웨어 개발자에 의해서 구축해야 할 시스템의 첫 번째 묘사로서, 시스템이 무엇을 하는지에 대한 기능 요구사항과 소프트웨어 개발자의 실제적인 제약사항에는 어떤 것들이 있는지에 대한 비기능 요구사항들을 포함해야 한다.

2.1절에서는 기존의 컴포넌트기반 소프트웨어 개발 프로세스들의 명세 활동들에 대하여 살펴보고, 2.2절에서는 본 논문에서 사용하고 있는 정형 방법과 정형 명세를 소개하고 정형 방법에서 사용하고 있는 정형 명세 언어 Z의 문법에 대하여 정리하였다.

2.1 기존 프로세스의 컴포넌트 명세 활동

2.1.1 Catalysis

Catalysis[1]는 UML, the OMG, RM-ODP를 포함하는 표준위에 생성된 객체와 프레임워크를 가진 컴포넌트 기반 개발을 위한 지원을 제공하는 방법론이다. 컴포넌트를 설계하는 과정은 크게 영역모델링, 시스템 모델링, 인터페이스 정의, 컴포넌트 인터페이스 명세, 사용자 인터페이스 설계의 5단계로 이루어진다(그림 1).

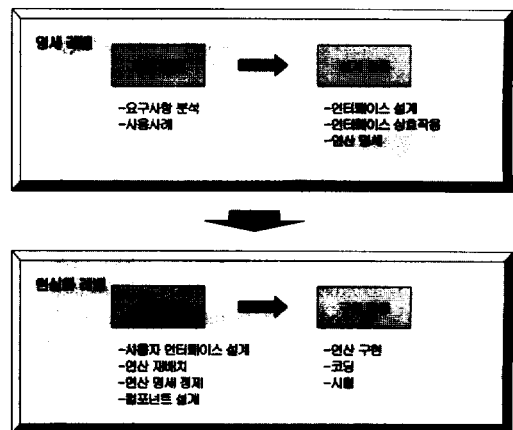


(그림 1) Catalysis 프로세스의 주요 활동

이중 컴포넌트 인터페이스 명세에서는 영역자원, 네트워크 구조, 보안등을 고려하여 컴포넌트의 단위를 결정한다. 결정된 컴포넌트에 대하여 지원되는 인터페이스들을 결정하여 할당한다. 그러나, 명세의 정확성과 검증을 실시하지 않고 있다.

2.1.2 CBD/e

CBD/e[2]는 Catalysis를 기반으로 하고 자체적으로 컴포넌트 모델링 프로세스를 포함하고 있고 여러 가지 도구에 의해서 지원된다. CBD/e는 2개의 부분으로 이루어진다. 하나는 조직이 자신들의 인력자원으로 자신들의 CBD 업무를 구현하기 위해서 따라야 하는 CBD/e 프로세스 그 자체이고, 다른 하나는 조직이 소프트웨어 컴포넌트를 분석하고 구현하기 위해서 따를 수 있는 분석과 설계 방법론이다. CBD/e 프로세스는 심사, 계획, 교육, 하부구조, 실행, 개선으로 구성되고, CBD/e 분석과 설계방법론은 UML 표기법과 다이어그램을 사용하여 시스템을 두 레벨로 설계한다(그림 2).

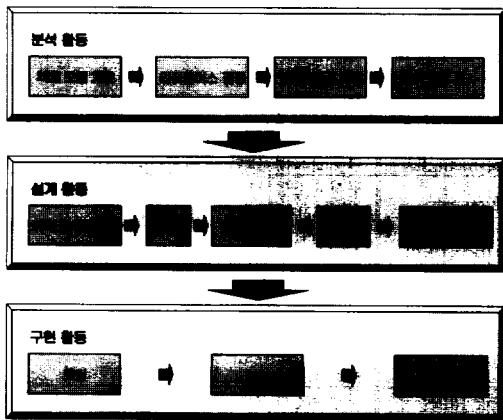


(그림 2) CBD/e 분석과 설계 프로세스

이중 설계에서는 시스템 요구사항을 지원하는 인터페이스와, 인터페이스들 사이에 발생하는 상호작용이 명세된다. 연산 명세에서는 각각의 연산에 존재하는 선조건과 후조건을 명세한다. 이러한 명세는 단순한 검사를 통해서 정제되어 질 수 있다.

2.1.3 Fusion

Fusion[3]은 요구사항 정의로부터 구현에 이르는 현존하는 Booch, Rumbaugh, 그리고 다른 방법들과 Objectory를 통합하고 확장한 것이다. Fusion은 분석, 설계, 구현의 3단계로 구성되고 요구분석 단계를 가지고 있지 않다(그림 3). 분석활동에서는 시스템의 행위가 정의되고, 설계활동에서는 시스템 연산이 상호작용하는 객체들, 클래스들 사이의 참조, 상속 관계, 클래스 속성, 클래스 연산에 의해서 구현되는 방법을 보여준다. 구현활동에서는 상속, 참조, 클래스 속



(그림 3) Fusion 방법론

성들이 프로그래밍 언어에 의해서 구현된다.

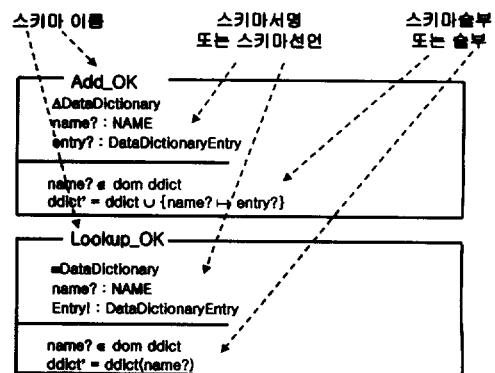
이중 분석활동에서는 문제영역과 그들 사이의 관계 개념이 설명된 객체 모델을 개발하고, 시스템 연산의 집합인 시스템 인터페이스를 정의하고, 일련의 상호작용을 보여주는 생명주기 모델과 연산 모델을 정의한다. Fusion은 모든 가능한 시나리오에 의한 검사(check)를 통해서 분석 모델을 검사한다.

2.2 정형 방법과 정형 명세 언어 Z

정형 방법(Formal Methods)은 대부분 수학에 존재하는 기본적인 개념들을 기초로 하여 소프트웨어의 품질을 제어하고 보증하기 위한 방법을 제공한다[4,6]. 즉, 정형 방법의 사용은 그것에 의해서 생산된 소프트웨어의 정확성에 있어서 신뢰성을 크게 증진시킬 수 있다. 그 중에서도 정형 명세 활동은 명세의 정확성과 일관성 그리고 만족해야 하는 특성들을 보유하고 있는지를 검증하고, 이것이 사용자의 요

구사항을 충족시키는지를 확인함으로써 요구사항 검증으로의 피드백과 설계와 구현으로의 진행에 있어서 기본이 되고 가장 중요한 활동이다.

정형 명세 언어 Z[7-9]는 표준 집합론에 기반하고 시스템의 특성을 표현하기 위해서 수학적 개념을 확장시킨 정형표기법이다. 그러므로, 시스템은 집합과 집합들간의 관계로서 설계되고 Z 스키마를 사용하여 구조화된다. Z 스키마는 시스템의 상태와 상태에 대한 연산이 정의된다(그림 4). 또한, 시스템의 제약사항을 표현함으로써 명세의 정형적인 부분뿐 아니라, 비정형적인 요구분석으로부터 정형 명세를 찾는 것을 도와준다.

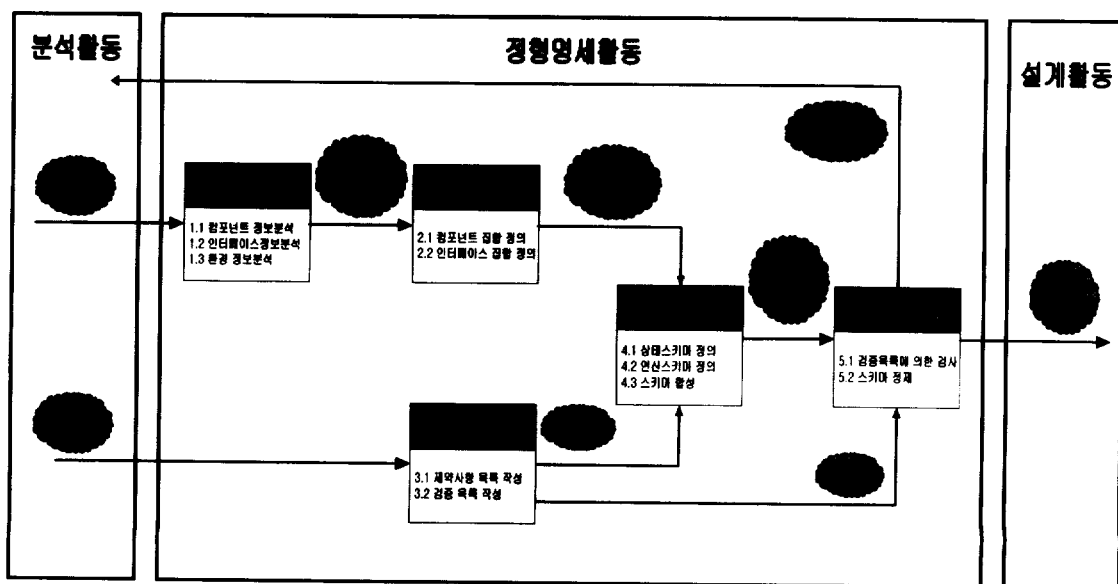


(그림 4) Z 스키마의 예

3. Z를 이용한 컴포넌트 정형 명세 활동

3.1 제안 명세 활동

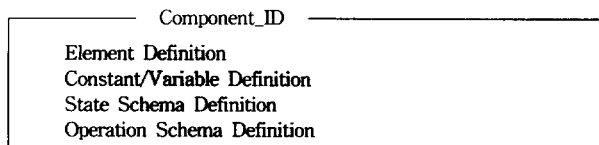
제안하는 컴포넌트 정형 명세 활동은 분석활동의 산출물인 요구사항 정의를 입력으로 하여 설계활동의 기초가 되



(그림 5) Z를 이용한 컴포넌트 정형 명세 방법

는 검증된 컴포넌트 명세를 산출하는 5개의 태스크로 구성된다. 이 태스크들은 12개의 서브태스크와 10개의 산출물을 생성한다(그림 5). 정형 명세 활동에서 실시하는 최초의 태스크는 컴포넌트를 명세하기 위해서 필요한 정보를 분석하는 명세 정보 분석 태스크와 스키마정의에서 필요한 제약사항과 검증을 위한 검증 목록을 작성하는 명세 조건 분석 태스크이다. 이들 태스크들은 순서를 가지지 않으며 병행하여 실시할 수 있다. 명세정보가 분석되면 컴포넌트 명세를 위해 필요한 집합/타입을 정의하는 태스크를 수행한다. 그리고, 집합/타입 정의 태스크와 명세 조건 분석 태스크의 정보를 바탕으로 컴포넌트의 상태와 기능을 Z 스키마로 정의하는 스키마 정의 태스크를 수행한다. 이렇게 정의된 스키마들이 합성되고, 합성된 스키마는 명세 조건 분석 태스크에서 제시한 제약사항 목록과 검증목록을 검사하여 명세 검증을 실시한후 정제가 요구되는 컴포넌트 명세의 경우는 분석활동으로 피드백을 제공하고 검증된 컴포넌트 명세는 설계활동의 기초가 된다. 각 태스크의 세부사항은 <표 1>, <표 3>, <표 4>, <표 7>, <표 8>과 같다.

컴포넌트 명세 스키마는 Element Definition(요소 정의부), Constant/Variable Definition(상수/변수 정의부), State Schema Definition(상태스키마 정의부), Operation Schema Definition(연산스키마 정의부)로 구성된다(그림 6).



(그림 6) 컴포넌트 명세 스키마

<표 1> 태스크 1.0 명세 정보 분석

태스크명	1.0 명세정보분석
정 의	요구사항정보로부터 컴포넌트 명세를 위해서 필요한 컴포넌트, 인터페이스, 환경정보를 식별
서브태스크	1.1 컴포넌트 정보분석 1.2 인터페이스 정보분석 1.3 환경 정보분석
산 출 물	컴포넌트 정보(Component_ID, Element) 인터페이스 정보(Link_Rules) 환경 정보(Component Framework)

요소 정의부에서는 컴포넌트의 구성요소들 즉 객체나 다른 컴포넌트들의 집합을 정의하고, 상수/변수 정의부에서는 컴포넌트 내부에서 필요한 상수와 변수를 정의하고, 상태스키마 정의부에서는 컴포넌트의 초기상태를 정의하고, 연산스키마 정의부에서는 컴포넌트의 요소들 사이의 관계 즉 인터페이스를 포함한 컴포넌트의 상호작용과 상태의 변화를 정의한다.

태스크 1.0 명세 정보 분석에서는 컴포넌트 이름, 컴포넌

트의 구성요소, 내부 및 외부 연결 규칙과 같은 기본적인 정보를 식별해야 한다<표 2>.

<표 2> 컴포넌트 명세에 필요한 기본정보

Component_ID	컴포넌트에 대한 유일한 이름
Element	구성요소들(객체 또는 컴포넌트)의 집합 =(COMPONENT, OBJECT)
COMPONENT	컴포넌트들의 집합 =(Component_ID Component_ID∈COMPONENT)
OBJECT	어떤 기능을 수행하는 최소 수행 단위 =class
INTERFACE	구성요소들과 그들과의 관계 =(Element, Link_Rules)
Link_Rules	구성요소들 사이의 내부 및 외부 연결 규칙

<표 3> 태스크 2.0 집합/타입 정의

태스크명	2.0 집합/타입 정의
정 의	컴포넌트 명세를 위해서 필요한 기본적인 구성요소 집합과 상수/변수를 정의
서브태스크	2.1 구성요소 집합 정의 2.2 상수/변수 정의
산 출 물	구성요소 집합 상수/변수 목록

<표 4> 태스크 3.0 명세 조건 분석

태스크명	3.0 명세조건분석
정 의	컴포넌트의 제약사항을 파악하여 제약사항 목록을 작성하고, 컴포넌트가 수행해야 할 기능들을 식별하여 검증목록을 작성
서브태스크	3.1 제약사항 목록 작성 3.2 검증 목록 작성
산 출 물	제약사항 목록 검증 목록

태스크 2.0 집합/타입 정의에서는 컴포넌트를 구성하는 구성요소들의 집합과 컴포넌트에서 필요한 상수 및 변수를 정의한다. 구성요소 집합과 상수는 Z에서 제공하는 집합 표기법을 이용하여 쉽게 정의할 수 있다(그림 7).



(그림 7) Stack의 변수 정의의 예

태스크 3.0 명세 조건 분석에서는 컴포넌트의 내부 연결 규칙과 외부 연결 규칙을 포함하여 제약사항 목록을 작성해야 하고, 컴포넌트 검증을 위한 검증 목록을 작성해야 한다<표 5>, <표 6>.

<표 5> 제약사항 목록 양식의 예

컴포넌트이름	컴포넌트 이름
제약사항	컴포넌트 내부 연결 규칙 컴포넌트 외부 연결 규칙
검증항목	요구되는 인터페이스 요청하는 인터페이스

<표 6> 검증 목록 양식의 예

검증 항목	요구분석 활동	명세 활동
요구되는 인터페이스 명세		
요청하는 인터페이스 명세		
제약사항 명세		
집합/타입 명세의 정확성		
초기상태스키마 명세		
올바른 연산스키마 명세		
예외처리 연산스키마 명세		

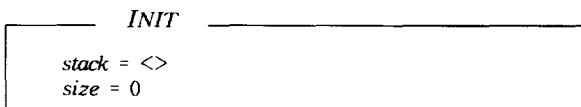
<표 7> 태스크 4.0 스키마 정의

태스크명	4.0 스키마 정의
정 의	태스크 1.0, 태스크 2.0, 태스크 3.0에서 식별된 정보를 이용하여 컴포넌트의 자료구조상태를 상태스키마로 정의하고, 컴포넌트가 수행해야 할 기능들을 연산스키마로 정의
서 브 태 스크	4.1 컴포넌트 상태스키마 정의 4.2 컴포넌트 연산스키마 정의 4.3 컴포넌트 스키마 합성
산 출 물	컴포넌트 상태스키마 올바른 연산스키마 예외처리 연산스키마

<표 8> 태스크 5.0 명세 검증

태스크명	5.0 명세 검증
정 의	컴포넌트 명세 검증
서 브 태 스크	5.1 검증목록에 의한 검사 5.2 스키마 정제
산 출 물	정제가 요구되는 컴포넌트 명세 검증된 컴포넌트 명세

태스크 4.1 컴포넌트 상태스키마 정의에서는 컴포넌트 각각에 대한 초기 상태스키마를 정의하여야 한다(그림 8).

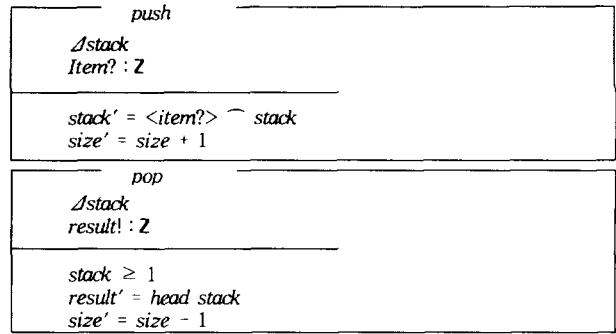


(그림 8) Stack의 초기 상태스키마의 예

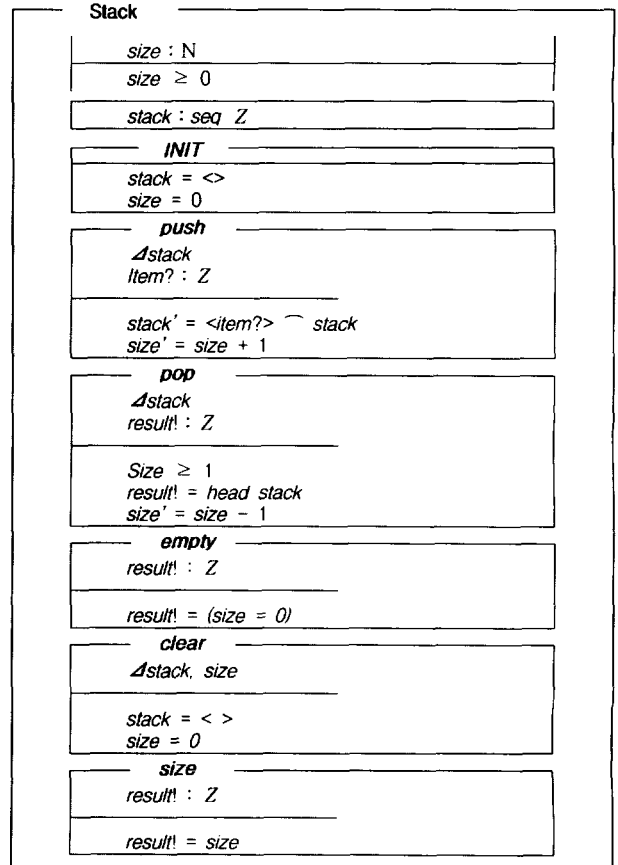
태스크 4.2 컴포넌트 연산스키마 정의에서는 내부 연결 규칙과 외부 연결 규칙의 식별에 따라 각각의 컴포넌트 연산스키마를 정의한다. 컴포넌트 내부 연결 규칙이 식별되면 컴포넌트의 상호작용과 관련된 인터페이스를 올바른 연산스키마와 예외처리 연산스키마 각각으로 정의한다. 외부 연결 규칙이 식별되면 컴포넌트들의 집합 사이의 상호작용을 명세하는 것으로 올바른 연산스키마와 예외처리 연산스키마 각각을 정의해야 한다(그림 9).

태스크 4.3 스키마 합성에서는 주어진 집합들과 상태스키마와 연산스키마를 합성하여 컴포넌트 명세를 완성한다(그림 10).

태스크 5.0 명세 검증에서는 명세 조건 분석 태스크에서 작성된 제약사항이 스키마 정의 태스크에서 명세된 스키마의 정확성 검사를 위해서 명세 조건 분석 태스크에서 작성



(그림 9) Stack의 push와 pop 연산스키마의 예



(그림 10) 상태스키마와 연산스키마의 합성의 예

한 검증 목록을 이용한다. 그리고, 컴포넌트 명세의 일관성은 전역정의의 일관성, 상태모델의 일관성, 연산의 일관성에 대하여 검증하고 자료와 연산에 대한 정제를 수행한다. 이러한 검증을 통해서 만약 정제가 요구되는 컴포넌트 명세가 발견되면 분석활동으로의 피드백을 제공하고, 그렇지 않은 경우는 검증된 컴포넌트 명세를 설계활동의 기초로 이용할 수 있다.

3.2 제안 활동의 비교

컴포넌트기반 소프트웨어 개발 프로세스의 명세활동은

정량적으로 비교하기 어렵기 때문에 각 프로세스의 명세 활동들을 명세서기, 명세도구, 품질평가, 그리고 특징 등을 정성적으로 비교하였다<표 9>. 기존의 컴포넌트 개발 프로세스의 명세활동에서는 다이어그램의 형태로 구문적인 측면은 잘 정의할 수 있지만, 의미적인 측면은 자연어로 기술함으로써 모호성을 가지고 있고 컴포넌트 품질에 대한 검증이 컴포넌트가 통합된 후에 가능하였지만, 제안 명세 활동은 수학적 모델링을 통한 정형 명세를 이용하여 구문적인 측면과 의미적인 측면을 정확하게 명세할 수 있을뿐 아니라 명세단계에서 컴포넌트 품질에 대해서 검증할 수 있는 장점이 있다.

<표 9> 명세 활동의 정성적 비교

프로세스 항목	Catalysis	CBD/e	Fusion	제안활동
명세서기	시스템명세	설계	분석	명세
명세도구	영역모델 형모델	사용사례	객체모델 인터페이스 모델	Z
품질평가	정제	정제	검사	검증
기능요구	○	○	○	○
비기능요구	×	×	×	○
특징	자연어 다이어그램	자연어 다이어그램	자연어 다이어그램	Z

4. 결론 및 향후연구방향

본 논문에서는 컴포넌트기반 소프트웨어 개발 프로세스의 초기활동, 즉 명세활동에서 명세의 정확성과 명세의 검증을 통하여 컴포넌트의 품질을 높일 수 있도록 분석력과 논리성이 검증된 정형 명세 언어 Z를 이용한 컴포넌트 정형 명세 활동을 제안하였다. 제안 활동은 명세의 정확성과 일관성 그리고 만족해야 하는 특성들을 보유하고 있는지를 검증하고, 명세가 사용자의 요구사항을 충족시키는지를 확인하기 위해서 구체적인 10개의 산출물과 12개의 서브태스크를 포함하여 5개의 태스크로 구성된다.

제안 활동은 개발초기에 명세의 정확성을 검증하여 품질 높은 컴포넌트 개발을 보증함으로써 생산된 컴포넌트의 신뢰성을 크게 증진시킬 수 있다. 또한, 비정형적인 사용자의 요구사항인 제약사항이 Z 스키마의 슬부에 명시됨으로써 스키마 정확성 검사를 통하여 사용자의 요구사항 정의를 정확하게 명세할 수 있다. 그리고, 정형 명세를 이용함으로써 명세이후의 설계, 구현, 시험 활동들의 비용을 크게 줄임으로써 전체 개발 비용을 절감할 수 있다. 그러나, 명세의 정확성과 일관성 검증을 위한 비용은 정형 명세를 이용하지 않는 프로세스보다 많이 든다. 그래서, 컴포넌트 정형 명세 활동을 지원해주는 도구의 개발과 개발 프로세스의 다른 활동들에서도 정형 방법을 적용할 수 있는 연구가 필요하다.

참고 문헌

- [1] D. F. D'Souza, A. C. Wills, 'Objects, Components, and Frameworks with UML', Addison-Wesley, 1999.
- [2] K. McInnis, "An Overview of CBD/e," <http://www.cbd-hq.com/>.
- [3] Coleman, "The Fusion method," <http://www.wis.cs.utwente.nl:8080/dmrg/ODOC/oodoc/oo-11.html>
- [4] B. Ratcliff, 'Introducing specification using z: A Practical Case Study Approach', McGraw-Hill, 1994.
- [5] I. Sommerville, 'Software Engineering', fifth edition, Addison-Wesley, 1998.
- [6] K. Lano, 'Formal Object-Oriented Development', Springer, 1995.
- [7] A. Harry, 'Formal Methods Fact File: VDM and Z', John Wiley & Sons, pp.173-258, 1996.
- [8] D. C. Ince, 'An Introduction to Discrete Mathematics, Formal System Specification, and Z', Clarendon Press, 1992.
- [9] D. Sheppard, 'An introduction to formal specification with Z and VDM', McGraw-Hill, 1995.
- [10] 장종표, 김병기, "Z를 이용한 컴포넌트 명세 전략 및 방법", 한국정보처리학회 춘계학술대회, 2000.
- [11] 장종표, 문성준, 정대영, 이상준, 김병기, "Z를 이용한 컴포넌트 상호작용 명세", 산·학·연 소프트웨어공학 기술학술대회 논문집, pp.324-329, 2000.
- [12] 장종표, 이상준, 김병기, "Z를 이용한 컴포넌트 명세 활동", 한국정보과학회 호남·제주지부 학술발표논문집, pp.28-35, 2000.



장종표

e-mail : jppang@chonnam.ac.kr
 1992년 전남대학교 전산통계학과(이학사)
 1997년 전남대학교 전산통계학과(이학석사)
 1997년~현재 전남대학교 전산학과 박사과정
 전남대 정보통신연구소 연구원
 관심분야 : SE, Formal Methods, CBSE, SPICE



이상준

e-mail : sjlee@tiger.seonam.ac.kr
 1991년 전남대학교 전산통계학과(이학사)
 1993년 전남대학교 전산통계학과(이학석사)
 1999년 전남대학교 전산통계학과(이학박사)
 1995년~현재 서남대학교 컴퓨터영상정보통신학부 조교수
 관심분야 : 소프트웨어공학, 분산 객체 시스템, 소프트웨어 품질평가



김병기

e-mail : bgkim@chonnam.ac.kr
 1978년 전남대학교 수학과(이학사)
 1980년 전남대학교 수학과(이학석사)
 2000년 전북대학교 수학과(이학박사)
 1981년~현재 전남대학교 컴퓨터 정보학부 교수
 관심분야 : SE, CBSE, 객체지향 시스템