실시간 속성 중심의 자바가상머신 명세도구의 설계 및 구현

고종원*, 송영재*
*경희대학교 컴퓨터공학과

e-mail:tavy95@khu.ac.kr

A Tools for Specification of Real-time Property centric Java Virtual Machine Components

Jong-Won Ko*, Young-Jae Song*
*Dept of Computer Engineering, Kyung-Hee University

요 약

임베디드 시스템에 탑재될 자바가상머신을 기능별 모듈로 컴포넌트화 하여 이식될 플랫폼의 요구사항에 맞추어 재구성한다면 컴포넌트 기술의 장점인 소프트웨어의 재사용과 재구성을 통한 빠르고 신뢰성있는 자바가상머신 개발이 가능하다. 이러한 자바가상머신 컴포넌트 재구성을 위한 컴포넌트 모델의 정의 및 명세를 위한 지원도구가 필요하며, 명세도구의 지원 하에 보다 시각적인 컴포넌트 구성 및 각 명세요소 정의를 바탕으로 한 자바가상머신 컴포넌트의 재구성에 요구되는 여러 제약조건이나 각 컴포넌트 간의 관계정의 등이 명세 되어질 수 있다. 또한 임베디드 시스템이 가지는 실시간 속성에 대해서 정의하여 이를 명세도구를 통해서 속성명세 및 설계 시에 예측성을 지원하기 위한 WCET 계산이나 우선순위 정의 등을 명세할 수 있다. 따라서, 본 논문에서는 명세방법을 제안하고 실시간 속성 중심의 자바가상머신 컴포넌트 명세도구를 설계하고 구현하였으며, 기존에 제안된 내장형 실시간 컴포넌트 개발도구와 비교하였다.

1. 서론

[1]임베디드 시스템 환경이 차세대 정보통신 산업의 주력으로 주목을 받으면서 이동성을 가지는 내장형 시스템에서 요구되는 실시간성, 이동성, 이식성, 경량화 등의 지원을 해결할 관련기술이 요구되고 있다. 또한 플랫폼 독립성과 이식성을 충실히 지원하는 환경을 제공하는 자바가상머신이 이동 내장형 시스템에서 여러 가지 응용 소프트웨어를 지원하는 핵심역할을 수행하게 되면서 자바가상머신에 대한 연구가 진행되고 있다. 임베디드 시스템에 탑재될 자바가상머신을 컴포넌트화 하여 이식될 플랫폼의 요구사항에 맞추어 재구성한다면 빠르고 신뢰성있는 자바가상머신의 개발이 가능해진다. 기존의 VEST나 MetaH 와 같은 실시간 임베디드 시스템을 위한 개발환경은 이미 존재하지만 이러한 개발환경을 자바가상머신이 탑재될 플랫폼을 목표로 한 자바가상머신 컴포넌트 재구성에 적용하기에는 설계단계에서 이루어지는 명세정의부터 제약과 한계가 따른다. 따라서 본 논문에서는 자바가상머신 컴포넌트 재구성을 위한 명세도구의 필요성을 인식하고, 임베디드 시스템에서 요구되는 실시간 속성에 대한 정의와 각 컴포넌트 간 관계정의를 바탕으로 이를 지원하는 명세도구의 설계와 구현을 하였다. 또한 실시간 속성 명세를 통해 설계 시에 WCET 계산에 대한 예측성 지원 및 기존 개발환경에서의 명세도구의 비교하여 그 장단점을 분석하였으며, 향후 명세도구를 좀 더 진화시켜 신뢰성 있는 컴포넌트 재구성을 지원하기 위한 분석기능과 탑재될 플랫폼의 요구사항과 관련해서 최적화 기법에 필요한 측정 메트릭 설계에 대한 연구가 진행 중이다.

1) 본 연구는 한국과학재단 목적기초연구(과제번호 : R01-2001-000-00357-0) 지원으로 수행되었음.

2. 관련연구

2.1 컴포넌트 기반 자바가상머신

본 논문에서의 제안에 앞서 이미 연구된 컴포넌트 기반 자바가상머신은 내장형 실시간 경량 자바가상머신에 대해 컴포넌트로 설계하고, 각 기능별로 세부 설계된 JVM 컴포넌트는 컴포넌트 기반 내장형 실시간 시스템을 위한 컴포넌트 모델인 PBO 모델을 기반으로 구성하였다. 아래 [표1] 은 이들 컴포넌트의 기능을 요약한 것이다. 자바가상머신은 클래스 로딩을 담당하는 클래스 로더, 클래스 검증기, 암호화와 인증 및 전자서명을 담당하는 보

안 관리자, 실행엔진, 가베지 콜렉터, 실시간 및 비실시간 자바 쓰레드 와 자바가상머신의 컴포넌트 수행을 제어하고 관리하는 스케줄러 및 통신과 입출력을 담당하는 컴포넌트를 주요 컴포넌트로 가지며, 서브 컴포넌트로써 RMA 나 EDF 와 같은 스케줄링 알고리즘 컴포넌트나 다양한 쓰레기 수집 알고리즘을 가진 메모리 관리 컴포넌트, 암호화 및 인증과 관련된 기능을 가진 컴포넌트 등이 있다.

이렇게 자바가상머신에 대한 기능별로 구성된 주요컴포넌트와 서브 컴포넌트들은 연구에서 지향하는 개발 툴셋의 라이브러리를 통해서 지원이 되며, 자바가상머신이 탑재되는 임베디드 시스템에서 요구하는 시간속성이나 기타 메모리 제약에 따른 경량화 문제, 자바가상머신에서 지원하는 서비스 및 성능에 관한 요구사항 등을 고려해서 다시 빠르고 신뢰성 있는 재구성이 가능해진다.

[표1. 자바가상머신 세부컴포넌트의 기능]

주요컴포넌트	세부컴포넌트	기 능
Garbage Collector Component	Generation GC	세대별 쓰레기 수집 알고리즘을 적용한 메모리관리 컴포넌트
	Mark&Sweep GC	마크&스윕 쓰레기 수집 알고리즘을 적용한 메모리 관리 컴포넌트
	Incremental GC	순차적 쓰레기 수집 알고리즘을 적용한 메모리관리 컴포넌트
	Compacting GC	단순 수거 쓰레기 수집 알고리즘을 적용한 메모리 관리 컴포넌트
Eventhandler Component	Synchronous	이벤트가 발생한 시점에 이벤트를 먼저 처리하는 동기화 이벤트 관리기법을 사용하는 이벤트 관리자 컴포넌트
	Asynchronous	이벤트의 발생시점과 상관없이 이벤트 핸들러가 스케줄러에 의해 스케줄 되어 동작하는 이벤트 관리자 컴포넌트
Communi-cation Component	WAP	무선 프로토콜인 WAP 을 지원하기 위한 컴포넌트
	HTTP	HTTP 프로토콜을 지원하기 위한 컴포넌트
Execution Engine Component	Interpreter	실행엔진 컴포넌트
	JIT Compiler	
Scheduler Component	RMA	Rate Monotonic 알고리즘을 적용한 스케줄러 컴포넌트
	EDF	earlist deadline first 알고리즘을 적용한 스케줄러 컴포넌트
Output Component	Speaker	내장형 환경에서의 일반적인 출력장치를 지원하는 컴포넌트
	touchscreen	
Input Component	Keyboard	내장형 환경에서의 일반적인 입력장치를 지원하는 컴포넌트
	Mouse	
	touchpen	
	touchscreen	
	Mic	
Security Component	Encryption	암호화와 관련된 기능을 담당하는 컴포넌트
	Authentication	인증과 관련된 기능을 담당하는 컴포넌트
	digital signature	전자서명과 관련된 기능을 담당하는 컴포넌트
Verifier Compenent	Class Verifier	클래스 검증을 담당하는 컴포넌트
Class loader Component	Class loader	자바가상머신의 클래스 로딩을 담당하는 컴포넌트

2.2 실시간 컴포넌트 명세

실시간 컴포넌트의 명세연구는 주로 시간제약과 이벤트흐름에 대한 처리를 위주로 진행되어왔다[2]. 실시간 컴포넌트에 대한 명세는 상태변화도나 스테이트 차트를 통한 시스템의 제어 흐름이나 사건 흐름도를 통해 이벤트 발생순서와 객체제어에 대한 내용을 표현하였으며, ROOM 모델링의 경우 비 기능적 제약조건에 대한 지원이 미흡하였다. 또한 기존 컴포넌트 기반 임베디드 실시간 시스템 개발환경에서의 실시간 컴포넌트 명세는 인터페이스 기술이 부족하다든지 프로세스를 기반으로 하여 구조명세에 대한 지원이 미흡하였다. RUBUS 시스템에 와서는 선점형 및 우선순위 스케줄링을 지원하면서 end-to-end deadline 계산이나 WCET를 명세하기 위해 시간속성 등을 태스크 단위의 컴포넌트 내부에 표현했다. 하지만, 합성 컴포넌트를 생성하기 위한 지원이 제공되지 않은 단점이 있다. UML의 확장 메카니즘을 이용한 Rose RT 에서의 실시간 컴포넌트 명세는 컴포넌트간의 구조적인 명세 및 UML에서의 스테이트 다이어그램을 이용한 상태 기계 표기로써 행동명세를 나타낸다.

2.3 컴포넌트 명세요소 정의

라이브러리로부터 자바가상머신 컴포넌트를 선택하여 내장형 시스템에 탑재할 자바가상머신을 개발하고자 할 때, 컴포넌트의 기능 및 컴포넌트간의 관계를 나타내기 위해 필요한 속성 등에 대한 정의는 아래와 같다.[5]

1) 정의 1. 컴포넌트 A는 컴포넌트에 의해서 제공되는 서비스 Function(F), 컴포넌트 실행을 제어하는 Task(T), In/Out port(I,O), 컴포넌트 파라미터(P), 컴포넌트의 상태를 나타내는 State(S), 시간속성을 정의하기 위한 Temporal property(TP), 제약조건(C)의 일곱 개의 튜플을 가진다.

2) 정의 2. 컴포넌트 A의 Temporal property(TP)는 시간에 대한 정확성이 중요한 RT System에서 요구되는 시간특성을 정의하기 위한 속성인 Frequency, Priority, Execution Time, Deadline 들로 컴포넌트의 실시간 속성을 나타낸다.

3) 정의 3. 컴포넌트 A의 Constraint(C)는 임베디드 시스템이 가지는 다양한 제약조건인 Memory, Size, Power 등을 기술할 때 사용한다.

4) 정의 4. 컴포넌트 A1 와 A2 간의 관계는 다음 다섯 가지의 관계로 정의한다.

가. Data Flow Connection

: 컴포넌트의 In/Out port를 통해서 전달되는 데이터 흐름

나. Dependency: 컴포넌트 간의 의존관계 명시

다. Precedence: 컴포넌트의 수행순서 정의

라. Mutual exclusion(MUTEX)

: 컴포넌트의 수행 중에 쌍으로 동시에 수행될 수 없는 관계

마. Association: 컴포넌트 간의 연관관계

5) 정의 5. Pre/Post condition을 통해서 컴포넌트 사이의 포트 연결 관계를 기술한다.

정의 1,2,3,4,5 를 통해서 설명한 것과 같이 자바가상머신 컴포넌트에 대한 명세를 위한 요소로서 컴포넌트가 가지게 되는 기능적인 속성에서부터 Input , Output port를 통한 인터페이스 명세, 또한 컴포넌트 간의 의존관계에 대한 명세를 비롯하여 데이터 흐름이나 구조명세 및 실시간 속성 기술과 수행순서에 대한 관계들을 정의한다. 이와 같은 명세 요소 정의를 통해서 기존의 임베디드 실시간 시스템 개발 명세도구에서의 명세차원에서의 단점을 개선하고 자바가상머신 컴포넌트 명세를 위해 최적화된 명세기법을 제공한다.

3. 자바가상머신 컴포넌트 명세연구

자바가상머신 컴포넌트 명세는 크게 컴포넌트에 대한 구조명세와 실시간 시스템에 대한 요구사항을 만족시키기 위한 실시간 속성에 대한 명세로 나누어 명세한다.

구조명세는 컴포넌트 구조를 쉽게 알아볼 수 있게 컴포넌트의

계층적 명세를 통해서 컴포넌트 레벨 다이어그램에서부터 상세 컴포넌트 레벨 다이어그램으로 명세하며, 주로 컴포넌트에 대한 기능적인 속성과 컴포넌트 사이의 관계를 나타낸다.
또한 실시간 속성 명세는 실시간 시스템에서 중요한 시간속성을 만족하기 위해 각 컴포넌트를 Task 로 고려해서 컴포넌트의 수행시간 및 WCET 와 우선순위 등의 시간속성을 명세한다.

3.1 구조명세

자바가상머신 컴포넌트 구조 명세는 크게 컴포넌트 레벨 다이어그램, 컴포넌트 인터페이스 다이어그램, 그리고 상세 컴포넌트 레벨 다이어그램으로 구분하여 명세를 한다. 컴포넌트 레벨 다이어그램은 앞서 설계된 주요 컴포넌트와 세부 컴포넌트들을 라이브러리로부터 선택하여 컴포넌트들의 조립으로 자바가상머신을 표현하고 이들 컴포넌트간의 의존관계를 나타내며, 컴포넌트 인터페이스 다이어그램은 각 컴포넌트의 포트 정의 및 컴포넌트간의 관계를 구체적으로 명시한다. 또한, 상세컴포넌트 레벨 다이어그램에서는 컴포넌트 내부와 컴포넌트가 가지는 속성 및 포트가 어떻게 구성되어 있는지를 나타낸다.
컴포넌트 레벨 다이어그램에서는 자바가상머신을 구성하기 위해 라이브러리에서 제공되는 가상머신 구성 컴포넌트를 선택하여 다이어그램으로 표현한다.
자바가상머신 컴포넌트의 인터페이스 명세는 컴포넌트 내부의 메소드 정의와 입력포트, 출력포트의 메소드 정의를 통해서 인터페이스를 명세한다.

3.2 실시간 속성 명세

자바가상머신 컴포넌트는 실시간 시스템을 지원하기 위한 시간속성을 가지며, 이러한 실시간 속성은 이미 앞선 명세요소 정의를 통해서 Deadline 이나 Execution Time, Priority, Frequency 등으로 그 속성을 정의했었다.
그 속성 정의를 살펴보면 아래와 같다.

가. Priority : 컴포넌트의 수행순서 정의
나. Frequency : 컴포넌트의 수행주기
다. Execution Time : 컴포넌트의 수행시간
라. Deadline : 컴포넌트의 수행이 완료되어야 할 시간

컴포넌트에서의 시간 요구사항을 명세한 속성을 바탕으로 컴포넌트 간의 Execution Time 에 대한 Deadline 계산이나 컴포넌트의 수행순서, 또는 포트와 포트사이의 데이터 연결 등을 명세할 수 있다. 구체적인 예로 End-to-end deadline 에 대해서 첫 번째 컴포넌트의 실행 시작시간부터 마지막 컴포넌트의 실행 종료시간까지의 최대시간을 다음과 같이 정의할 수가 있다.

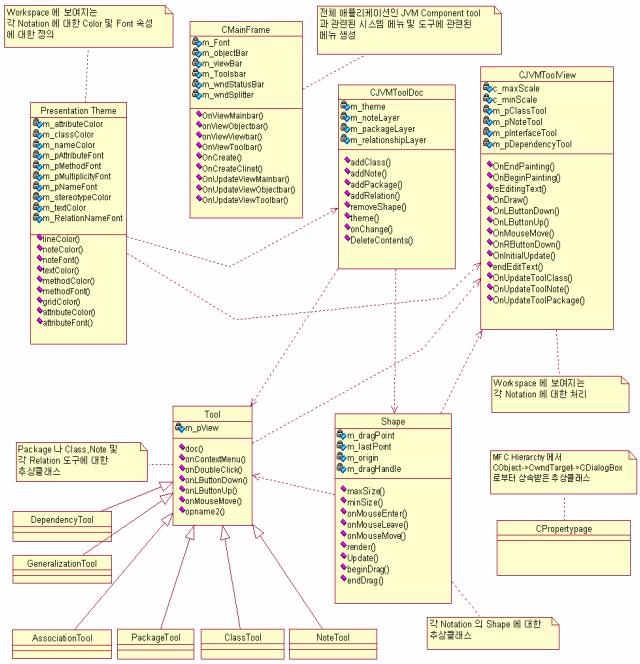
$$e2e = \mathrm{Max}(\text{execution time}(C2), \text{execution time}(C3) - \text{Earliest start time}(C1))$$

이러한 실시간 속성은 각 컴포넌트는 적어도 하나 이상의 Task 를 수행하므로 명세도구를 통해서 해당 컴포넌트에 명세함으로써 자바가상머신 컴포넌트 설계 시에 실시간속성에 대한 예측성이 지원가능하다.

4. 컴포넌트 명세도구의 설계 및 구현

4.1 자바가상머신 컴포넌트 명세도구의 설계

본 장에서는 자바가상머신 컴포넌트를 명세하기 위한 명세요소 정의를 바탕으로 이를 지원하는 명세도구를 구현하기 위해 UML로 명세도구의 설계를 위한 시스템 모델링을 하였다.
아래의 [그림 1] 은 명세도구의 중요한 클래스 간의 관계 및 속성과 메소드를 나타낸 다이어그램이다.
PresentationTheme 클래스는 화면에 보여주는 다이어그램에서의 Line 의 두께나 모양, 색깔과 텍스트의 폰트에 관한 속성, 또는 각 Notation 에 대한 색상에 대한 속성을 정의한 클래스이다. Tool 과 Shape 클래스, 그리고 CPropertypage 클래스는 추상클래스로 각 명세요소, 예를 들어, 컴포넌트나 패키지(Package), 주석(Note), 그리고 포트(Port), Association(연관관계), Dependency(의존관계) 등의 표기를 나타내는 도구 바를 클릭해서 워크스페이스(Workspace) 상에서 나타낼 때 처리해주는 클래스들을 생성할 때 추상클래스로부터 상속을 받는다.

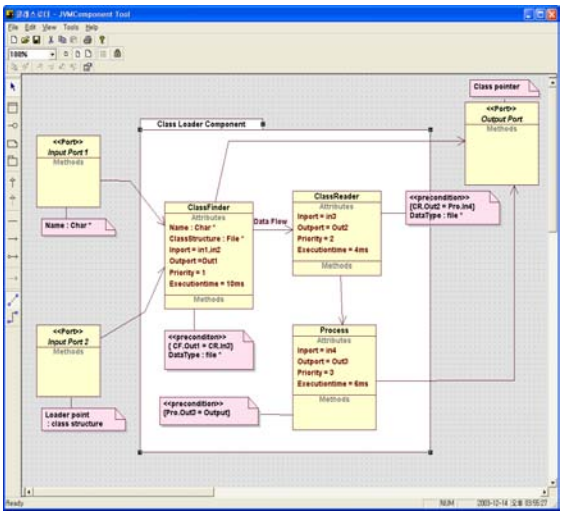


[그림 1] 명세도구에 대한 드로윙 관련 클래스다이어그램

4.2 자바가상머신 컴포넌트 명세도구의 구현

앞 장에서 살펴본 바와 같이 자바가상머신 컴포넌트 명세도구의 설계를 바탕으로 VC++ 6.0 개발환경을 이용해서 자바가상머신 컴포넌트 명세도구를 구현하였다.
다음 [그림 2] 은 앞서 명세요소 정의를 통해서 이미 UML을 통해서 명세하였던 자바가상머신의 클래스 로더에 대한 명세를 제안한 명세도구를 이용해서 나타낸 화면이다.
클래스로더는 상위 패키지의 개념으로 정의를 했으며, 그 패키지 내에 내부컴포넌트들로 ClassFinder, ClassReader, Process 와 같은 내부 컴포넌트를 가지고 클래스로더 패키지는 두 개의 입력 포트와 한 개의 출력포트를 가진다.
또한 클래스 로더 내부에 각 컴포넌트 간의 관계는 먼저 각 컴포넌트와 입출력포트 간에 주고받는 정보에 대한 관계인 데이터

흐름(Data Flow)에 대해 나타내고 있다. 주석(Comment) 을 통해서 각 포트에서 획득된 정보의 타입을 명세한다.
그리고 사후조건인 Precondition, Postcondition 에 대한 제약을 두어서 각 컴포넌트 포트별 연결 관계나 포트에서의 데이터타입을 제약조건으로 명세한다.
마찬가지로 [그림 2] 에서 보여주는 바와 같이 각 포트나 각 컴포넌트에서의 메소드 정의를 통해서 컴포넌트와 포트에서 제공하는 인터페이스에 대한 명세가 메소드를 통해서 나타난다.



[그림 2] 구현한 명세도구를 이용한 명세 예

5. 기존 명세도구와의 비교분석

본 논문에서 제안한 자바가상머신 컴포넌트 명세도구는 그 관련도구로써 임베디드 실시간 시스템 개발환경인 VEST, MetaH, CIP 개발환경과 실시간 시스템 모델링 도구인 RoseRT 와 Telelogic 으로 같이 그 비교항목을 살펴보면서 제안한 자바가상머신 컴포넌트 명세도구와 함께 각 도구별 차이점과 개선사항, 그리고 장점과 단점에 대해서 비교분석한다.
구조명세의 지원여부에 대해서 살펴보면, 제안한 명세도구를 포함한 전반적으로 모든 도구가 지원을 하고 있으며, 다만 CIP 도구의 경우 프로세스를 기반으로 하고 있기 때문에 프로세스의 구조명세가 가능하며, 전체 시스템에 대한 정적인 구조명세는 다소 부족하다.
또한, 실시간 속성에 대한 예측성과 관련해서 기타 다른 도구들도 제안한 명세도구와 함께 실시간 속성에 대한 예측성을 저마다 크고 작게 지원을 하고 있으며, VEST 나 MetaH 의 경우 개발환경의 분석도구에서 예측성에 대한 분석을 지원한다. 하지만 미흡하거나 분석도구에서의 구체적인 분석방안이 마련되지 못한 경우도 있다.
또한 논문에서 제안한 명세도구의 단점일 수도 있는 시스템의 행동명세 지원과 코드 자동생성 기능은 제안한 명세도구에서는 현재 지원하지 못하지만 이미 기존의 비교 명세도구는 이러한 두 기능을 지원한다.
제안한 명세도구는 자바가상머신 컴포넌트 설계 시에 컴포넌트의 기능적인 속성과 각 컴포넌트 간 관계 정의를 통한 구조 명세와 시간 속성을 만족시킬 수 있는 실시간 속성 명세가 가능하다. 하지만 다른 관련도구에 비해서 행위명세가 부족한 부분과 코드생성이 지원되지 않는다.

[표 2] 기존의 다른 명세도구와의 비교평가

비교도구 / 비교항목	임베디드 실시간 시스템 개발환경			실시간 시스템 모델링 도구		제안한 명세도구
	VEST	MetaH	CIP	RoseRT	Telelogic	JVM Tool
명세 요소 단위	Component	Component	Process	Class	Class	Component
명세 기반 모델	VEST컴포넌트모델	MetaH 언어	CIP컴포넌트모델	RT-UML	RT-UML	PBO 모델
구조명세 지원 여부	가능	가능	부분가능	가능	가능	가능
실시간 속성 예측성 지원	부분지원	분석도구에서 부분지원	지원안함	지원안함	지원	지원
행동명세 지원 여부	지원	지원	지원	지원	지원	부분지원
코드자동생성 기능지원	지원	지원	지원	지원	지원	지원못함
장 점	컴포넌트 간 의존성 검사 지원	H/W,S/W 컴포넌트 모델링 및 다양한 분석도구 지원	프로세스 구조와 행위명세에 적합	높은 상용성	높은 상용성	JVM 컴포넌트 명세에 유리
명세상의 한계점(단점)	인터페이스 명세가 지원안되며, 구체적인 분석방안이 없음	항공전자공학분야에 국한	시스템 구조명세 미흡	모델링개념이 복잡, JVM 명세 적용에 한계	모델링개념이 복잡, JVM 명세 적용에 한계	행동명세의 부분지원 및 코드생성기능 없음

6. 결론

본 논문에서는 이미 선행연구로써 진행된 개선된 PBO 모델에 기반하여 설계된 자바가상머신 컴포넌트들과 명세요소 정의를 바탕으로 자바가상머신 컴포넌트들 간의 구조명세는 물론 의존관계나 포트를 통한 컴포넌트 사이의 데이터 흐름, 수행순서 제어 등을 쉽게 명세할 수 있으며, 컴포넌트 조립 시에 기능적, 비 기능적 속성에 대한 설계과정에서의 정적인 분석이 가능하다. 또한 자바가상머신 컴포넌트 설계의 기본모델이 된 PBO모델이 가진 시간속성에 대한 표현이 부족한 단점을 개선하여 컴포넌트 명세에 필요한 여러 실시간 속성들을 이용해서 deadline 이나 컴포넌트 수행 간의 WCET 계산으로 컴포넌트 설계단계에서의 실시간 컴포넌트에 대한 예측성을 지원가능하게 하였다. 하지만 본 논문에서 제안한 명세도구의 단점인 행위명세에 대한 지원이 미흡한 부분과 코드자동생성 기능, 그리고 실시간 속성에 대한 분석기능에 대한 지원 등을 고려해서 향후 계속적인 관련연구를 바탕으로 명세도구를 계속적으로 개선시켜 나갈 계획이다.

참고문헌

[1] David B. Stewart, "Designing Software Component for Real-Time Applications", *2001 Embedded Systems Conference*, April. 2001
[2] Damir Isovic, Christer Norstrom, "Components in Real-Time Systems", Malardalen University, 2000
[3] Magnus Larsson, Anders Wall, Christer Norstorm, Ivica Crnkovic "Using Prediction-Enabled Technologies for Embedded Product line Architecture", *ICSE workshop on 5th ICSE Workshop on Component-based Software Engineering*, May.2002.
[4] Jose Javier Dolado, "A Validation of the Component-based Method for Software Size Estimation", IEEE Transaction on Software Engineering, Vol 26, No 10, 2000
[5] 고종원,서영준,이승룡,송영재, “컴포넌트 기반 자바가상머신 명세에 관한 연구”, 한국정보처리학회 춘계학술대회, 2003