

# UML 2.0을 사용한 모델 기반의 임베디드 소프트웨어 소모 전력 분석을 위한 MARTE Profile의 확장

## (Extension of MARTE Profile for Model-based Power Consumption Analysis of Embedded Software with UML 2.0)

편 호 림<sup>†</sup>      김 종 필<sup>†</sup>      홍 장 의<sup>††</sup>  
(Ho-Rim Pyeon)      (Jong-Phil Kim)      (Jang-Eui Hong)

**요 약** 임베디드 시스템의 개발에서 저전력을 소모하는 소프트웨어 개발에 대한 요구가 증대하고 있다. 따라서 임베디드 소프트웨어 개발 과정에서 소모 전력에 대한 정량적인 예측을 가능하도록 하는 연구가 늘어나고 있는 추세이다. 기존의 소모 전력 분석은 소스 코드를 중심으로 이루어져왔으나 분석을 위한 노력 및 시간이 많이 요구된다는 단점으로 인하여, 소프트웨어 모델 기반의 소모 전력 분석 기법에 관심을 두는 추세이다. 본 논문은 UML 모델 기반의 임베디드 소프트웨어 모델링 과정에서 설계 모델을 이용한 소모 전력 분석에 주안점을 두었으며, 이를 위하여 OMG가 개발한 MARTE 프로파일을 확장하였다. 이러한 확장은 별도의 분석용 모델을 개발하지 않고, UML 다이어그램을 이용한 소모 전력 분석이 가능하도록 한다.

키워드 : 임베디드 소프트웨어, 모델기반 전력 분석, MARTE

**Abstract** The needs of low-power embedded software are being increased. Along with the needs, the studies to predict the power consumption of embedded software are also being increased. Although existing studies for power analysis have been performed in source code-based, these code-based analysis have some shortages of long analysis time and much feedback efforts. Recently some studies of power analysis based on software models are prompted. This paper describes on the model-based approach using UML diagrams in embedded software development process. Specially we focus on the extension of OMG's MARTE Profile to support model-based analysis. The MARTE extension gives the possibility of power analysis using just UML diagrams without any other analysis model in embedded software development.

Key words : Embedded software, UML models, Power analysis, MARTE

## 1. 서 론

임베디드 시스템이 다양한 응용 분야로 확산되면서, 내장되는 소프트웨어의 기능 또한 복잡해지고 있다. 이러한 상황에서 휴대성이 강조되는 임베디드 시스템의 경우는 저전력 소모의 요구가 증대하고 있으며, 이로 인하여 소모 전력이 적은 내장형 소프트웨어를 개발하기 위한 연구가 진행되고 있다.

저전력 소모에 대한 임베디드 소프트웨어 요구사항을 충족시키기 위해서는 소프트웨어 개발 과정에서 전력 소모량의 정량적인 예측이 가능해야 한다. 소프트웨어에 대한 소모 전력의 예측에 대한 연구들은 다양한 방법에 의해 진행되어 왔다[1,2]. 기존의 임베디드 소프트웨어에 대한 전력 분석은 명령어나 소스 코드 수준에서 이루어

· 본 논문은 2009년도 정부(교육과학기술부)의 재원으로 한국과학재단의 지원을 받아 수행된 연구임(No. R01-2008-000-20485-0)

† 학생회원 : 충북대학교 컴퓨터학과  
hrpyeon@selab.cbnu.ac.kr  
kimjp@selab.cbnu.ac.kr

†† 종신회원 : 충북대학교 컴퓨터공학 교수  
jehong@chungbuk.ac.kr

논문접수 : 2009년 8월 6일

심사완료 : 2010년 2월 12일

Copyright©2010 한국정보과학회 : 개인 목적이나 교육 목적인 경우, 이 저작물의 전체 또는 일부에 대한 복사본 혹은 디지털 사본의 제작을 허가합니다. 이 때, 사본은 상업적 수단으로 사용할 수 없으며 첫 페이지에 본 문구와 출처를 반드시 명시해야 합니다. 이 외의 목적으로 복제, 배포, 출판, 전송 등 모든 유형의 사용행위를 하는 경우에 대하여는 사전에 허가를 얻고 비용을 지불해야 합니다.

정보과학회논문지: 소프트웨어 및 응용 제37권 제4호(2010.4)

했으며[3-5], 최근에는 모델 기반의 전력 분석 기법들도 병행적으로 연구되고 있다[6-8].

기존의 명령어나 소스 코드 기반의 소모 전력 분석 기법들은 대상 시스템에 대한 가상 프로토타이핑(virtual prototyping)을 통해 소프트웨어의 소모 전력을 예측하였다. 이러한 방법을 통해 얻은 소모 전력의 예측치는 실제 시스템의 소모 전력과 큰 차이를 보이지 않았지만, 분석을 위해 소요되는 시간이 방대하고, 분석 결과를 피드백(feedback)하여 얻는 소모 전력 절감 효과가 미비하다는 단점이 있다. 이는 분석 단위가 명령어와 같은 하위 수준의 요소들을 이용하여 수행되기 때문이다. 또한 소스 코드가 생성된 이후에나 분석이 가능하기 때문에 피드백을 위한 노력이 많이 소요된다는 단점도 있다.

따라서 비록 실제 시스템의 소모 전력량과는 차이가 있지만, 코드 기반 분석보다 선행적으로 수행할 수 있는 모델 수준에서의 소모 전력 분석 기법이 활발히 연구되고 있다. 모델 기반의 분석은 다수의 명령어나 프로그램 문장이 추상화된 모델 요소를 중심으로 소모 전력을 분석하는 접근 방법이다. 임베디드 소프트웨어의 소스 코드가 개발되기 이전에 분석이 이루어지기 때문에 제작업에 대한 노력이 적고, 또한 전력을 절감하기 위한 단위가 소스 코드에 비해 상대적으로 크다. 이와 같이 모델 수준에서 임베디드 소프트웨어의 다양한 비기능적 특성, 예를 들면 실시간성, 성능, 스케줄 가능성 등과 같은 특성을 분석할 수 있도록 하는 틀이 마련되고 있는데 이것이 OMG가 제시하는 “UML Profile for MARTE (Modeling and Analysis of Real-Time and Embedded systems)”[9] - 줄여서, MARTE 프로파일 - 이다.

OMG는 실시간 시스템의 시간과 성능에 관한 요구사항을 수용하기 위한 “UML Profile for Schedulability, Performance, and Time”[10] - 줄여서, SPT 프로파일 - 에 임베디드 시스템의 다양한 비기능적인 요구사항을 모델링할 수 있도록 확장한 MARTE 프로파일을 제시하였다. 그러나 비록 MARTE 프로파일이 임베디드 시스템의 비기능적 요구사항을 표현하고 분석하는데 유용한 정보를 포함하고 있다 할지라도, UML로 작성한 소프트웨어 모델을 기반으로 전력 분석을 수행하기에는 분석에 필요한 정보들이 추가적으로 필요하다[1]. 따라서 본 연구에서는, UML 기반의 임베디드 소프트웨어 모델에 대한 전력 분석을 수행하기 위해 UML 모델에 전력 관련 속성을 표현하는 방법을 제안하고, 이를 지원할 수 있도록 MARTE 프로파일을 확장한다. 또한 예제 시스템 적용을 통해 확장된 프로파일의 적합성을 확인한다.

본 논문은 다음과 같이 구성되어 있다. 2장에서는 관련 연구로 모델 기반 전력 분석 연구와 실시간 임베디드 시스템의 모델링을 지원하는 MARTE 프로파일에

대하여 간략히 살펴본다. 3장에서는 임베디드 소프트웨어의 모델 기반 전력 분석을 위해 필요한 속성들을 도출하고, 이러한 속성들의 명세를 위해 MARTE 프로파일을 확장한다. 그리고 4장에서는 확장된 프로파일을 사용하여 모델 기반 전력 분석을 위한 예제 시스템 모델링 후 전력 분석에 어떻게 적용되는지를 확인한다. 마지막으로 5장에서는 향후 연구 내용에 관하여 기술하고, 본 연구 내용을 요약하며 결론을 짓는다.

## 2. 관련 연구

### 2.1 모델 기반 전력 분석

임베디드 소프트웨어의 모델 기반 전력 분석에 대한 연구는 아직 초보 단계에 있다. 소모 전력 감소를 위한 대부분의 연구가 시스템이나 실행 개념 측면에서 초점을 두고 있기 때문이다. 모델 기반 전력 분석에 대한 연구로는 Tan[6], Jun[7], 그리고 Yue[8]의 연구가 있다.

Tan은 임베디드 소프트웨어 아키텍처 레벨에서 전력 분석을 수행하기 위한 방법[6]을 제안하고 있다. 이 방법은 소프트웨어 아키텍처를 프로세스 상호간의 커뮤니케이션에 초점을 둔 SAG(software architecture graph)로 변환하고 에너지 매크로 모델을 이용하여 그래프에서의 상호작용에 대한 전력 소모량을 분석한다. Jun은 컴포넌트 기반의 임베디드 소프트웨어에서 전력 소모량을 모델링하고 분석하기 위한 방법[7]을 제안하고 있다. 인터페이스 오토마타를 확장하여 수행 시간과 에너지 소모 비율을 기술할 수 있는 에너지 인터페이스 오토마타를 정의하여 전력 분석에 활용하고 있다. Yue의 연구는 하드웨어 플랫폼에 독립적으로 모듈 단위의 에너지 모델을 정의하고 재사용하기 위하여 객체 지향 에너지 모델[8]을 제안하였다. 전력 소모량을 분석하기 위해 가상 명령어를 정의하고 이를 이용하여 설계 모델에 주해를 삽입한다. 에너지 모델은 가상 명령어의 에너지 소모 값과 매핑되어 해당 모듈에 대한 전력 소모량을 예측하는 것에 사용된다.

이와 같은 연구들이 갖는 특성은 대상 소프트웨어에 대한 소모 전력을 분석하기 위하여 SAG나 에너지 인터페이스 오토마타와 같은 별도의 모델을 개발하고, 이를 분석하기 위한 기반 구조를 함께 정의하고 개발해야 한다는 것이다. 또한 분석 결과가 개발 중인 소프트웨어의 산출물로 반영되기 위한 방법이 구체적으로 제시되지 못하고 있다. 그러나 본 연구에서는 대상 소프트웨어 개발 시 작성되는 UML 다이어그램을 이용하도록 함으로써, 별도의 추가 노력 없이 소모 전력 분석이 가능하도록 하는데 주안점이 있다.

위와 같이 소프트웨어의 소모 전력 분석기법을 제안하는 기존의 연구들이 수행되어 왔으나, UML 기반의 설계 모델을 이용한 소모 전력 분석기법에 관한 연구는

찾아보기 어렵다. 단지 기존에 UML SPT 프로파일을 확장하여 하드웨어 모델링을 지원하고자 하는 연구[11]와 MARTE 프로파일을 확장하여 소프트웨어 구성요소간의 의존성 분석을 위한 연구[12]는 있으나, 이는 단순히 하드웨어 아키텍처 모델링 및 소프트웨어 의존성 분석을 지원하기 위한 것으로써, 모델 기반의 소모 전력 분석을 지원하는 연구라고는 할 수 없다.

2.2 MARTE 프로파일

UML은 프로파일을 통해 특정 도메인이나 기술적인 영역에서 다양한 어플리케이션의 모델링을 지원한다. UML 프로파일은 어떤 특정 관심 도메인의 메타모델로써, 관심 도메인 고유의 모델 라이브러리를 포함한다. 그리고 이러한 모델 라이브러리의 표현을 위해 스테레오타입과 그와 연관된 태그 값을 사용한다. 이러한 특정 도메인 메타모델의 모든 요소는 UML 메타클래스의 스테레오타입으로 나타낸다[13].

OMG는 최근 실시간 임베디드 시스템 도메인을 위한 메타모델로써, MARTE 프로파일을 제시하였다. MARTE 프로파일은 기존의 실시간 시스템의 모델링을 위해 사용되던 SPT 프로파일을 대체함으로써, 스케줄 가능성, 성능, 그리고 실시간성 등의 명세와 함께 임베디드 시스템의 비기능적 속성들의 명세도 가능하도록 확장하였다 [14]. MARTE 프로파일은 그림 1과 같이 세 개의 주요 패키지와 하나의 부가 패키지로 구성된다[9].

- MARTE Foundations 패키지 : 실시간 임베디드 시스템의 모델 개발과 모델 기반 분석을 위해 공통으로 사용되는 기본 개념들을 정의한다. 실시간 임베디드 시스템의 비기능적 속성들에 관련된 개념(NFP 패키지)과 시간 관련 개념(Time 패키지), 플랫폼을 위한 일반적인 자원 모델(GRM 패키지), 자원의 할당 관련 개념(Alloc 패키지)이 이러한 범주에 속한다.

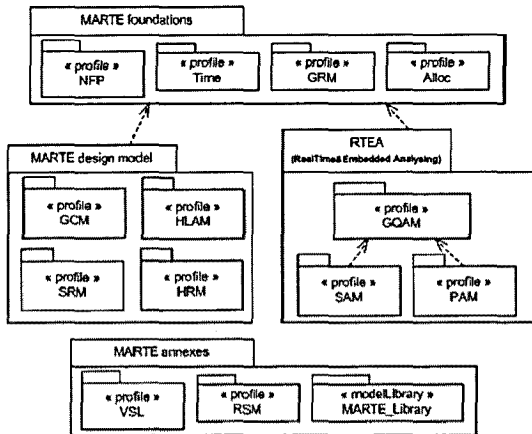


그림 1 MARTE 프로파일의 아키텍처

- MARTE Design Model 패키지 : 실시간 임베디드 시스템의 설계 모델과 관련한 개념들을 정의한다. 어플리케이션을 위한 일반 컴포넌트 모델(GCM), 상위 레벨의 어플리케이션 모델(HLM), 상세화된 자원 모델링을 위한 소프트웨어 자원 모델(SRM)과 하드웨어 자원 모델(HRM)이 이에 속한다.
- RTEA 패키지: 실시간 임베디드 시스템의 비기능적 속성들의 모델 기반 분석을 위한 개념들로 이루어져 있다. 기존의 SPT 프로파일에서 스케줄 가능성과 성능 분석을 위해 사용하던 스케줄 가능성 분석 모델(SAM), 성능 분석 모델(PAM)을 포함하고, 이 두 가지 모델에서 공통적으로 포함하는 내용을 기초로 하여 실시간 임베디드 시스템의 여러 비기능적 요구사항들의 모델 기반 분석을 위한 일반적인 정량적 분석 모델(GQAM)을 정의한다.
- MARTE annexes 패키지 : MARTE 프로파일에서 사용되는 개념들에 대한 부가적인 사항을 모아 놓은 패키지이다.

이와 같은 패키지들로 구성된 MARTE 프로파일을 사용하여 UML로 명세된 모델의 분석 절차는 그림 2와 같다.

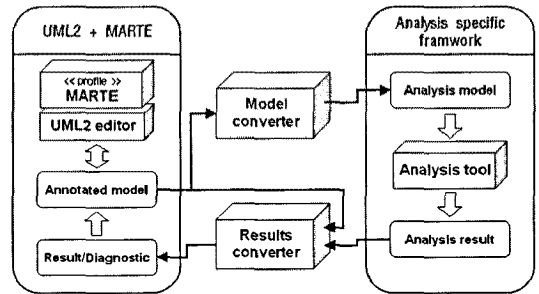


그림 2 모델 기반 분석 절차

MARTE 프로파일과 UML 2.x를 사용하여 모델링된 임베디드 소프트웨어 모델은 모델 컨버터를 통해 분석 가능한 모델로 변환되며, 변환된 모델은 적절한 분석 도구를 통해 정량적으로 분석된다. 분석 결과는 작성한 모델이 요구사항을 만족하는지 판단하기 위하여 사용되며, 만약 요구사항을 만족하지 않는 경우, UML로 작성한 모델은 수정되어야 한다.

3. MARTE 프로파일의 확장

UML을 사용한 임베디드 소프트웨어의 모델링에 있어서, 비록 MARTE 프로파일이 비기능적인 요구사항들의 명세를 통해 모델 기반의 정량적인 분석을 가능하게 하지만, 임베디드 소프트웨어의 모델에 전력 분석을 위한 속성들을 표현하기에는 현재 정의된 개념만으로는

충분하지 않다. 이는 그림 1에서와 같이 현재 MARTE 프로파일이 실시간 임베디드 시스템의 비기능적인 속성들에 관한 분석 모델을 추상화시켜 RTEA 패키지 내의 GQAM 패키지로 정의하고, 스케줄 가능성 분석을 위한 SAM 패키지, 그리고 성능 분석을 위한 PAM 패키지만을 특성화된 분석 모델로 정의하고 있기 때문이다. 따라서 본 연구에서는, 임베디드 소프트웨어의 모델 기반 전력 분석을 위해, 전력 소모 관련 속성들의 명세가 가능하도록 MARTE 프로파일을 확장하고자 한다.

3.1 전력 소모 인자

본 연구에서는 임베디드 소프트웨어에 대한 전력분석을 위한 기본적인 프레임워크를 그림 3과 같이 정의하였다. 소프트웨어의 동작을 통한 소모 전력의 분석은 소프트웨어 컴포넌트(프로세스 또는 태스크)가 하드웨어 아키텍처에 어떻게 할당되는지에 따라 소모 전력이 다르게 결정될 수 있기 때문이다.

그림 3과 같이 임베디드 소프트웨어 모델링은 응용 시스템 모델로부터 소프트웨어 아키텍처 모델을 작성하는 PIM(Platform Independent Model) 부분과 하드웨어 아키텍처 모델을 생성하고, 이를 소프트웨어 구성요소들과 매핑한 PSM(Platform Specific Model) 부분으로 구분하여 생각할 수 있다. 소프트웨어 소모 전력은 하드웨어 구성요소들의 동작에 의존적이기 때문에 소모 전력의 분석은 PSM의 결과를 기반으로 수행되어야 한다.

임베디드 소프트웨어의 소모 전력 분석은 그림 4와 같은 메타 모델에 의해 표현될 수 있다. PIM을 이루는 소프트웨어 모델 요소(SW model elements)들은 이들의 구현을 통한 명령어(Instruction) 수준의 패턴으로 정의할 수 있다. 명령어는 Load, Add, Store 등과 같은 기본 명령어(Primitive instruction) 집합과 wait(), fork() 등과 같은 함수 명령어(Sys\_function instruction)로 구분된다. 이들 명령어들은 피연산자의 타입에 따라서 소모되는 전력이 차이가 있기 때문에 int, char, double 등과 같은 피연산자(operand) 타입의 정보를 필요로 한다.

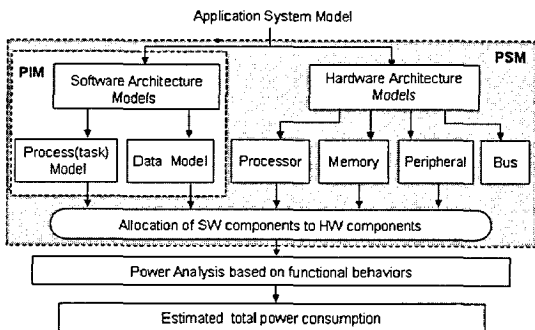


그림 3 임베디드 소프트웨어의 소모 전력 예측 프레임워크

이와 같이 명령어 패턴으로 표현되는 각각의 모델 요소들은 프로세스 또는 실행단위로 그룹핑(Element cluster)되어 물리적인 하드웨어 컴포넌트에 할당되어 실행된다. 실행 과정에서 소모되는 전력량은 하드웨어 요소에 의존적이며, 이들에 대한 정보는 표 1과 같이 MARTE의 HRM 패키지 내에 정의되어 있다.

또한 명령어를 구성하고 있는 피연산자의 타입과 이들에 대한 정보는 MARTE model library에 기본 데이터 타입으로 정의하고 있으며 Allocation에 대한 부분도 MARTE Alloc 패키지에서 정의하고 있다. 따라서 본 연구에서는 그림 4에 정의된 모델 요소 중에서 표 2와 같은 MARTE 확장 요소를 정의할 수 있다.

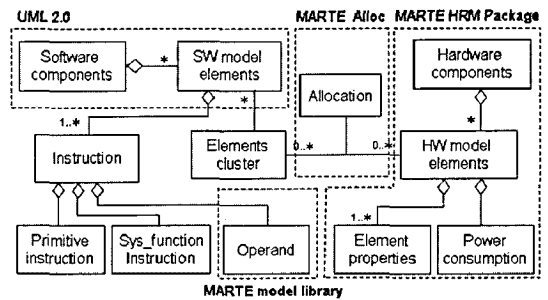


그림 4 소프트웨어 소모 전력 분석의 메타 모델

표 1 MARTE가 정의하는 하드웨어 컴포넌트 속성들

패키지명	클래스명	속성명
HW_Computing	HW_Processor	- op_frequencies - mips, nbPipeline, nbStages
HW_Storage	HW_Cache	- memorySize, Throughput, writePolicy - level, type, structure
HW_Storage	HW_RAM	- nbRows, nbColumns, wordSize
HW_Comm.	HW_Bus	- addressWidth, wordWidth,
HW_Device	HW_IO	- NOT DEFINED
HW_Timing	HW_Clock	- frequency
	HW_Timer	- nbCounter, counterWidth
HW_Power	HW_PowerDescriptor	- consumption, dissipation
	HW_PowerSupply	- suppliedPower
	HW_Battery	- capacity
	HW_CoolingSupply	- coolingPower

3.2 프로파일 확장 과정

3.2.1 GQAM 패키지 분석

앞서 언급한 바와 같이 MARTE 프로파일에서는 실시간 임베디드 시스템의 여러 가지 비기능적인 요구사항에 대한 분석 모델 구축을 돕기 위해 확장 가능한 GQAM을 정의하고 있으며, 세부적으로는 Schedulability와 Performance에 대한 분석 모델만을 정의하고 있다. 따라서 본 논문에서는 표 2에 나타난 확장 요소들을 표현 가능하도록 MARTE 프로파일의 GQAM을 확장한다.

MARTE 프로파일을 확장하기 전에 먼저 GQAM 패키지에 대하여 간단히 언급하고자 한다. 그림 5와 그림 6은 각각 GQAM 패키지와 MARTE 프로파일의 다른 패키지들과의 연관성, GQAM 도메인 모델의 최상위 패키지를 보여준다.

그림 5에서 보는 바와 같이 GQAM 패키지는 MARTE 프로파일의 다양한 개념들을 사용하여 분석 모델을 구축하는데 필요한 개념들을 정의하고 있으며, 이러한 개념의 구성에 필요한 용어들은 MARTE 프로파일을 구성하는 다른 패키지들과 마찬가지로 MARTE Library에서 정의하고 있다.

또한 GQAM은 이벤트에 의해 발생하는 행위를 Workload 개념을 기반으로 기술하도록 하며, 자원과 시간 제약 사항에 대한 부분을 정량적으로 명세 가능하도록 하고 있다. 이렇게 명세된 부분은 추후 분석 도구를 통해 정량적인 분석이 가능하도록 돕는다. 본 논문에서는 GQAM의 Workload 개념에 대한 내용만을 간략히 기술하고, 다른 부분들은 MARTE 프로파일의 명세서에 상세히 기술되어있기 때문에 생략한다[9].

GQAM에서 Workload는 시스템이 갖는 여러 가지 다른 상황들과 각각 연관된다. 각각의 Workload는 이벤트의 발생 흐름을 표현하는 Workload Event와 이벤트에 영향을 받아 수행되는 Behavior Scenario로 구성된다. Workload Event는 이벤트의 발생 패턴에 관한 속성을 포함하여 이벤트가 어떠한 패턴으로 발생하는지

표 2 MARTE 프로파일 확장에 필요한 요소

확장요소	세부 내용
Instruction	UML 모델 요소에 대한 classifier에 대응하는 명령어 식별자와 해당되는 명령어 패턴을 가짐. 즉, 모델 요소를 표현하는 Primitive ins.와 Sys_function ins.들로 구성.
Primitive instruction	일반적인 수준에서의 명령어 목록을 표현함.
Sys_function instruction	일반적인 수준(임베디드 운영체제)에서의 시스템 함수 명령어 목록을 표현함.
Element cluster	프로세스(태스크)를 구성하는 모델 요소들의 모임으로 하드웨어 요소로의 매핑을 위한 단위임.

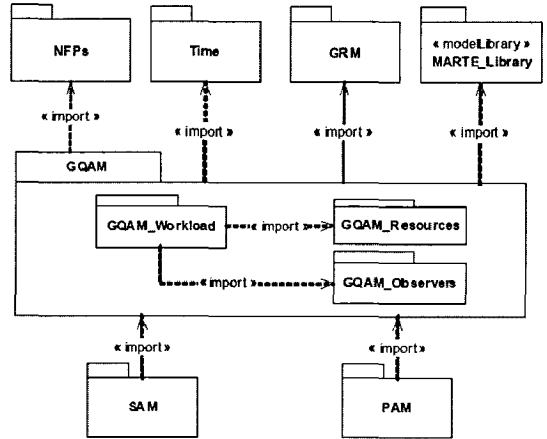


그림 5 GQAM 패키지와 다른 패키지들과의 연관성

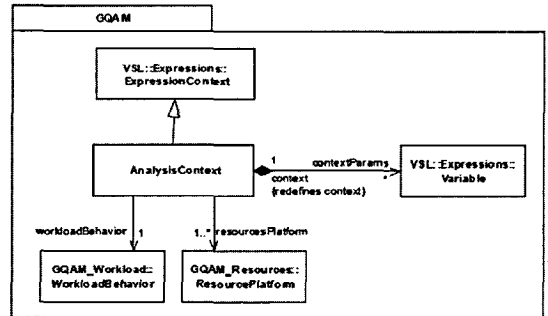


그림 6 GQAM 도메인 모델의 최상위 패키지

명세 가능하도록 정의되어 있다. Behavior Scenario는 세부적으로 Step들로 구성되어 각각의 Step을 통해 전체 시나리오를 표현할 수 있도록 정의되어 있다. 또한 Step에 여러 속성을 포함하여 분석에 필요한 여러 가지 정보를 표현할 수 있도록 돕는다. 이러한 GQAM의 Workload 개념에 관한 도메인 모델은 그림 7과 같다.

3.2.2 EAM 프로파일 개요

표 2에서 제시한 소프트웨어 소모 전력 분석을 위한 속성들은 앞서 설명한 GQAM을 확장하여 포함될 수 있으며, 확장된 MARTE 프로파일을 기반으로 전력 분석을 수행하기 위한 모델을 본 논문에서는 에너지 분석 모델(Energy Analysis Model, EAM)이라 정의하였다.

그림 8은 EAM의 상위 도메인 모델을 보여준다. GQAM에서 분석 시나리오 전반에 걸쳐 필요한 속성과 시나리오를 구성하는 행위 각각에 분석을 위해 필요한 속성을 명세하는 것과 같은 방법으로 전력 분석을 위한 모델을 구축하기 위해 EAM에서는 GQAM의 Analysis Context와 Workload 개념을 사용한다. 또한 전력 분석을 위해 필요한 속성에 대한 개념이 MARTE Library

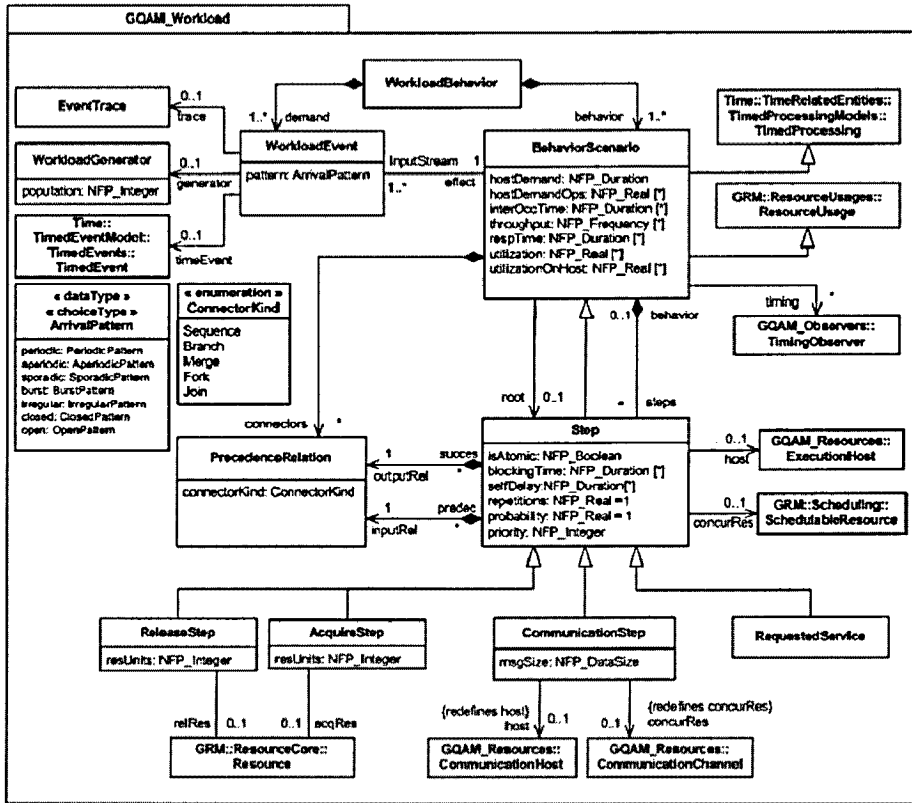


그림 7 GQAM\_Workload 패키지의 도메인 모델

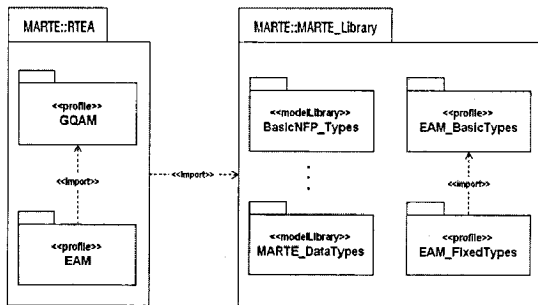


그림 8 EAM의 상위 도메인 모델

에 정의되어 있지 않은 경우, 라이브러리를 확장하여 새로운 개념을 추가하여야 한다. 때문에 전력 분석을 위한 확장 요소 Primitive instruction과 Sys\_function instruction을 보다 자세히 정의할 필요가 있다. 그림 8에 정의된 MARTE Library에 표 2에서 정의한 명령어를 반영하기 위해서는 표 3과 같은 확장된 명령어들을 정의할 수 있다.

표 3의 명령어들은 소프트웨어 명령어가 실행될 때, 하드웨어의 동작에 따른 소모 전력 특성을 나타내기 위해 하드웨어 플랫폼과의 종속성을 최대한 배제하여 일

표 3 확장 요소가 포함하는 명령어들

확장 요소	명령어
Primitive instruction	load, store, add, call divide, call modulo, mult, compare, convertType, branch, bitAND, bitNOT, bitOR, bitXOR, bitShiftLeft, bitShiftRight
	Process Manager fork(), waitpid(), wait(), signal()
Sys_function instruction	IPC msgsnd(), msgrcv(), msgget(), msgctl(), semget(), semctl(), semop(), pipe(), pipe open(), pipe write()
	File System File open(), File close(), File read(), File write()
	Memory Manager shmget(), semat(), shmdt(), shmctl()

반화한 형태로 정의된 명령어들이다. Primitive instruction은 임베디드 시스템 구성에 많이 사용되는 RISC 방식의 프로세서 중 하나인 StrongArm에서 제공하는 명령어를 참조하여 일반화한 형태이며, Sys\_function instruction은 Embedded Linux에서 제공하는 시스템 함수를 참고하여 일반화한 형태이다[15].

3.2.3 MARTE Library의 확장

MARTE Library에는 앞서 설명한 명령어에 대한 내용은 정의되어 있지 않다. 때문에 본 논문에서는 이러한 내용을 MARTE Library에 포함시키기 위해 전력 분석을 위해 필요한 확장 요소들을 EAM\_BasicTypes라는 이름으로 정의하였다. 그림 9는 EAM을 구축하는데 필요한 기본 명령어 타입을 정의한 MARTE model library인 MARTE\_Library::EAM\_BasicTypes를 보여준다.

또한 표 2에서 제시한 확장 요소 중에서 Instruction은 UML 모델 요소에 대한 classifier에 대응하는 명령어 식별자와 해당되는 명령어 패턴을 갖는다. 즉, 모델 요소를 표현하는 Primitive instruction과 Sys\_function instruction들로 구성된다. MARTE 프로파일은 기본적으로 모델 작성자가 예측 가능한 값을 정량적으로 명세하도록 구성되어 있지만, 사실상 UML 모델 요소가 어떠한 명령어로 구성되는지를 모델 작성자가 예측하기는 불가능하다. 때문에 UML 모델 요소와 명령어 간의 직접적인 관계 또한 MARTE model library에 정의되어야 할 필요가 있다.

그림 10은 UML 모델 요소와 Action Semantic을 구성하는 명령어 패턴을 정의한 MARTE model library인 MARTE\_Library::EAM\_FixedTypes이다. MARTE 프로파일은 고정적인 값을 명세할 방법을 제공하고 있지 않기 때문에 fixedType라는 스테레오 타입을 사용하여 확장하였다. 명령어 패턴은 UML 모델을 실제로 구현하여 생성된 소스 코드를 기반으로 결정된다[15]. 이렇게 MARTE model library를 확장하여 정의한 개념들은 EAM을 구축하기 위해 사용된다.

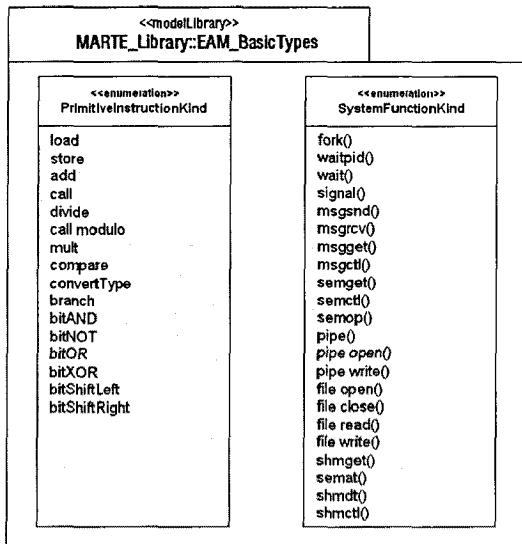


그림 9 MARTE\_Library::EAM\_BasicTypes

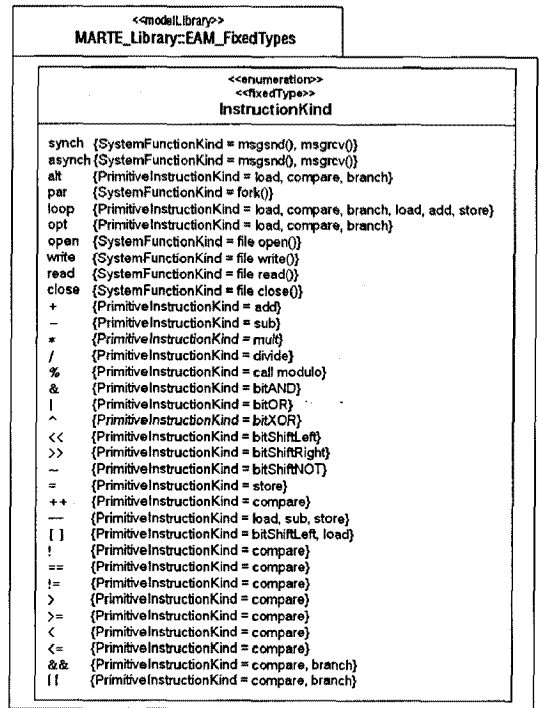


그림 10 MARTE\_Library::EAM\_FixedTypes

3.2.4 EAM 프로파일 명세

그림 11은 분석 시나리오의 각 부분을 구성하는 행위들에 전력 분석을 위한 속성의 명세를 위해 GQAM\_Workload를 확장한 결과이다. GQAM\_Workload의 CommunicationStep을 확장하여 EAStep을 정의함으로써 GQAM\_Workload에서 기본적으로 포함하고 있는 속성들과 더불어 EAM에서 전력 분석을 위해 필요한 속성 또한 모델에 명세할 수 있다.

확장된 결과를 UML 모델링에 적용하기 위해서는 UML 메타모델 또한 확장할 필요가 있다. 이는 EAM을 통해 UML 모델에 명세 가능하도록 확장한 내용들이 UML 모델링에서 어떻게 사용될 것인지를 나타내는 지표가 된다. 그림 12에서 EAM 표현을 위해 UML 메타모델을 확장한 결과를 보여준다.

그림 12에서 보이듯이 EAStep은 GQAM에서의 UML 메타모델 확장 방법과 마찬가지로, UML 메타클래스 BehavioralFeature에 스테레오타입의 형식으로 명세할 수 있도록 한다. 보다 자세한 속성 명세 방법은 4장에서 예제 시스템 모델링을 통해 알아보도록 한다.

3.2.5 Element Cluster의 확장

표 2에서 제시한 확장 요소 중 Element cluster는 UML 모델의 분할에 의해 생성되는 태스크의 도출 결과를 의미한다. UML 모델로부터 태스크를 도출하기 위

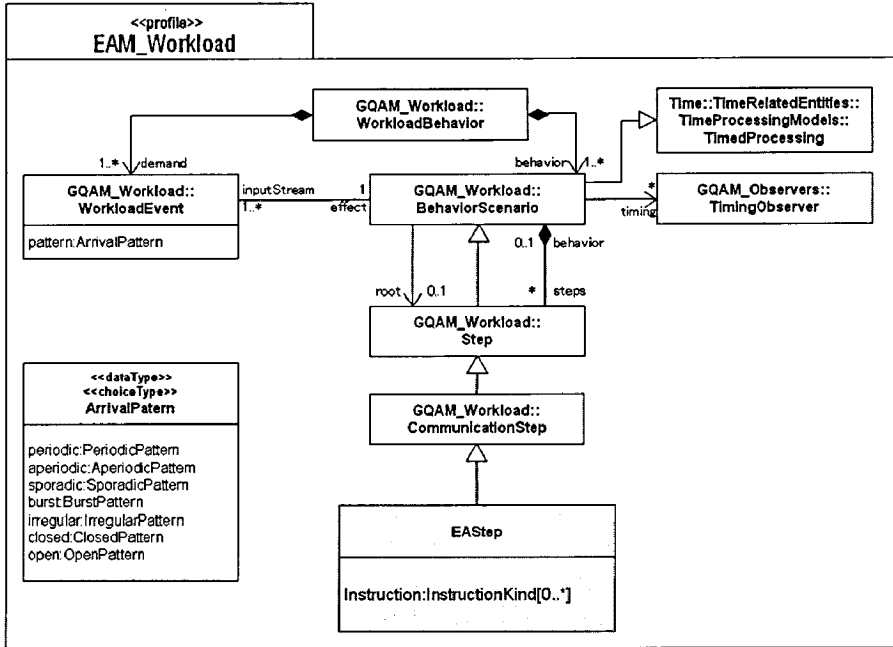


그림 11 EAM\_Workload 도메인 모델

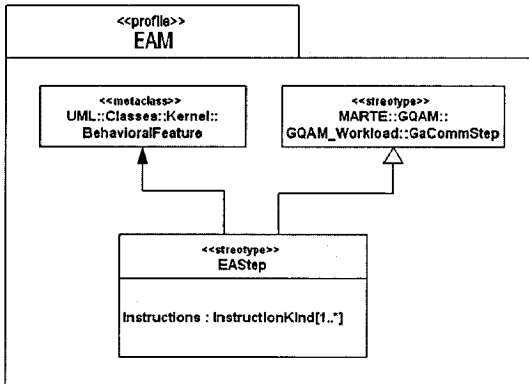


그림 12 EAM 표현을 위한 UML 메타모델 확장

해서는 UML 모델을 CFG(Control Flow Graph) 형태로 변환하고, 이 그래프를 응집력을 기준으로 분할한다. 분할된 그래프의 조각은 태스크로 고려되며, 이들은 하드웨어 아키텍처를 구성하는 요소에 할당된다[16]. 도출된 태스크에 어떠한 UML 모델 요소가 포함될 것지는 응용 모델에 따라 달라진다. 때문에 태스크는 UML 모델 요소에 대한 정보를 포함할 수 있어야 한다.

MARTE 프로파일에서는 HLAM(High-Level Application Modeling) 패키지를 통해 어플리케이션의 상위 레벨 모델링을 지원하며, DRM(Detailed Resource Modeling) 패키지를 사용하여 소프트웨어 리소스와 하드웨

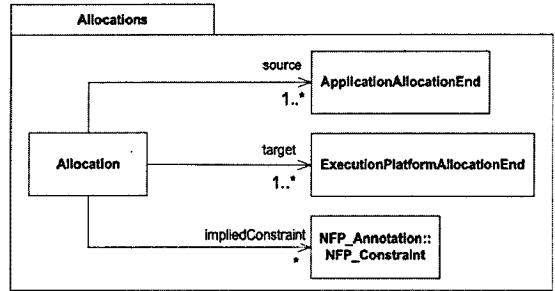


그림 13 MARTE 프로파일의 Allocation model

어 리소스의 상세한 모델링에 대한 개념들을 정의하고 있다. 이렇게 각각 어플리케이션 레벨과 하드웨어 플랫폼 레벨에서 모델링된 결과는 Alloc 패키지를 통하여 연관 관계가 정의된다. 즉, Alloc 패키지에서는 그림 13에서 보이는 바와 같이 어플리케이션 레벨에서의 태스크를 가용한 하드웨어 리소스에 할당하는 개념을 정의하고 있다. 그림에서 NFP\_Constraint는 time에 관한 제약사항으로써 어플리케이션 레벨의 logical time과 하드웨어 플랫폼 레벨의 physical time의 관계를 설명하기 위해 사용된다.

본 논문에서는 태스크에 UML 모델 요소의 정보를 포함시키기 위해 MARTE 프로파일의 HLAM 패키지를 확장한다. HLAM 패키지는 앞서 설명한 바와 같이 상위 레벨의 어플리케이션 모델링을 가능하게 한다. 하지



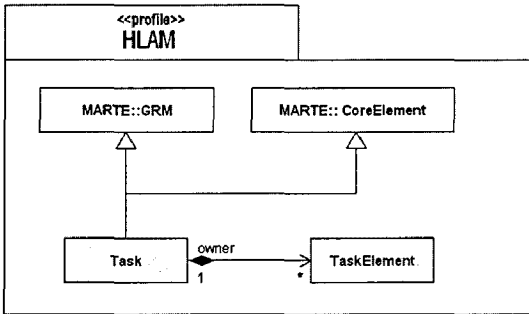


그림 14 Task 개념을 추가한 MARTE HLAM 패키지의 확장

만 현재 MARTE 프로파일에서는 실시간 임베디드 시스템의 concurrency 특성을 다루기 위해 HLAM 패키지를 활용하고 있다. 때문에 우리는 그림 14와 같이 HLAM 패키지에 Task라는 개념을 적용하여 확장하였다.

그림 14에서 보이듯이 Task는 TaskElement들로 구성되며, 여기서 TaskElement는 Task를 구성하는 UML 모델 요소들의 정보를 포함하기 위한 요소이다. UML 모델의 CFG 변환 후 그래프의 분할을 통해 구성되는 태스크들은 위에서 확장한 Task로 매핑될 수 있으며, 이러한 매핑 시 UML 모델 요소들에 대한 정보는 Task-Element로 유지된다. 이러한 UML 모델 요소들에 대한 정보는 전력 분석을 하는 과정에서 사용될 것이다.

#### 4. 예제 시스템에 대한 적용

##### 4.1 소프트웨어 모델링

그림 15는 전력 분석 목적으로 확장된 MARTE 프로파일인 EAM 프로파일을 휴대용 디지털 사운드 레코더의 음성 재생 기능 모델링에 적용한 것이다. 모델링에는 시스템 순차적인 행위를 통한 전력 분석을 수행하기 위해 시퀀스 다이어그램을 사용하였다.

그림 15와 같이 기본 모델에 EAM 프로파일을 이용하여 표 2에서 제시한 전력 분석에 필요한 확장 요소를 추가하고 부가적으로 MARTE 프로파일의 GQAM에서 제공하는 요소(prob, repetition)를 명세함으로써 보다 정밀한 전력 분석이 가능하도록 하였다.

##### 4.2 전력 분석에서의 적용

앞서 언급했던 사항들을 기반으로 모델 기반의 소모 전력 분석은 그림 16과 같은 과정을 거쳐 이루어진다.

그림 16에서 보는 바와 같이 UML 모델 요소들은 각 명령어 패턴에 의해 정의되며, 이들 명령어들은 에너지 라이브러리에 저장된 각 단위 명령어별 소모 전력량을 이용하여 소프트웨어의 소모 전력을 계산하게 된다. 또한 UML 모델로부터 추출된 태스크들은 모델 요소를 포함하고 있고, 모델 요소들은 명령어들로 표현이 가능하기 때문에 태스크가 갖는 파이프라인 스톱 확률이나 캐시 미스율 등을 명령어 수준에서 계산할 수 있다. 파이프라인 스톱 확률, 캐시 미스율과 같은 하드웨어에 의

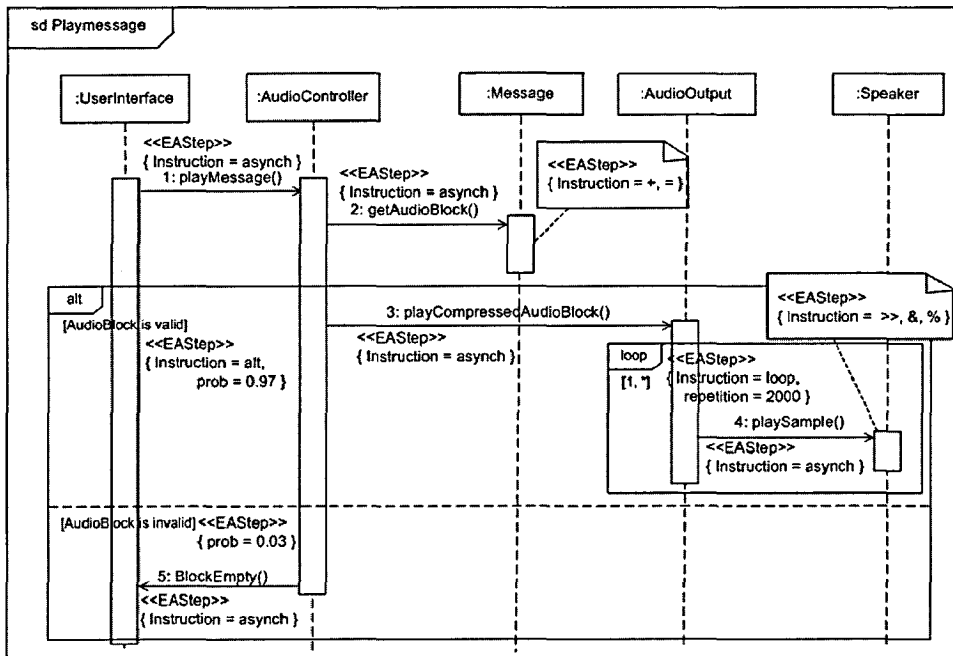


그림 15 EAM 프로파일이 적용된 예제 시스템

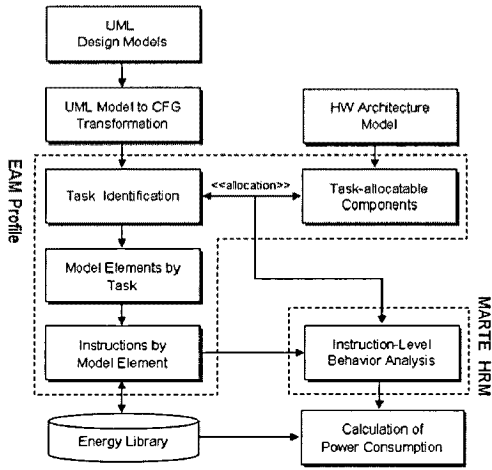


그림 16 모델 기반 소모 전력 분석 절차

존적인 정보들은 MARTE HRM을 이용한 하드웨어 상세 모델링을 통해 충분히 예측 가능하다. 분석하고자 하는 모델의 전체 태스크의 수를  $l$ , 각각의 태스크가 포함

하고 있는 모델 요소의 수를  $m$ , 각각의 모델 요소가 포함하고 있는 명령어의 수를  $n$ 이라 할 때, 위에서 설명한 정보들을 기반으로 해당 모델의 소모 전력  $P(M)$ 을 산출하는 최종의 계산은 다음 식에 의해 이루어진다.

$$P(M) = \sum_{t=1}^l \sum_{c=1}^m \sum_{i=1}^n (1 + S_i + C_i) \times J_{t,c,i} \quad (1)$$

식 (1)에서  $J$ 는 해당 명령어의 소모 전력을 의미하며,  $S_i$ 와  $C_i$ 는 각각 명령어의 파이프라인 스톱 확률과 메모리 캐시 미스율을 의미한다.

### 4.3 적용 결과

앞서 제시한 MARTE 프로파일의 확장이 소모 전력 분석 과정에서 어떠한 역할을 하는지를 좀 더 명확하게 나타내기 위하여 4.1절의 예제 시스템 모델과 에너지 라이브러리와의 매핑 과정을 보다 자세히 설명하고, 모델 기반의 소모 전력량 분석 결과를 제시함으로써, 확장된 MARTE 프로파일의 유용성을 확인한다. 그림 16에 나타난 에너지 라이브러리(Energy Library)는 소모 전력 분석을 위해 선행적으로 구축되어야 하는 분석 프레임워크의 구성요소로서, 특정 하드웨어 아키텍처 상에서 각

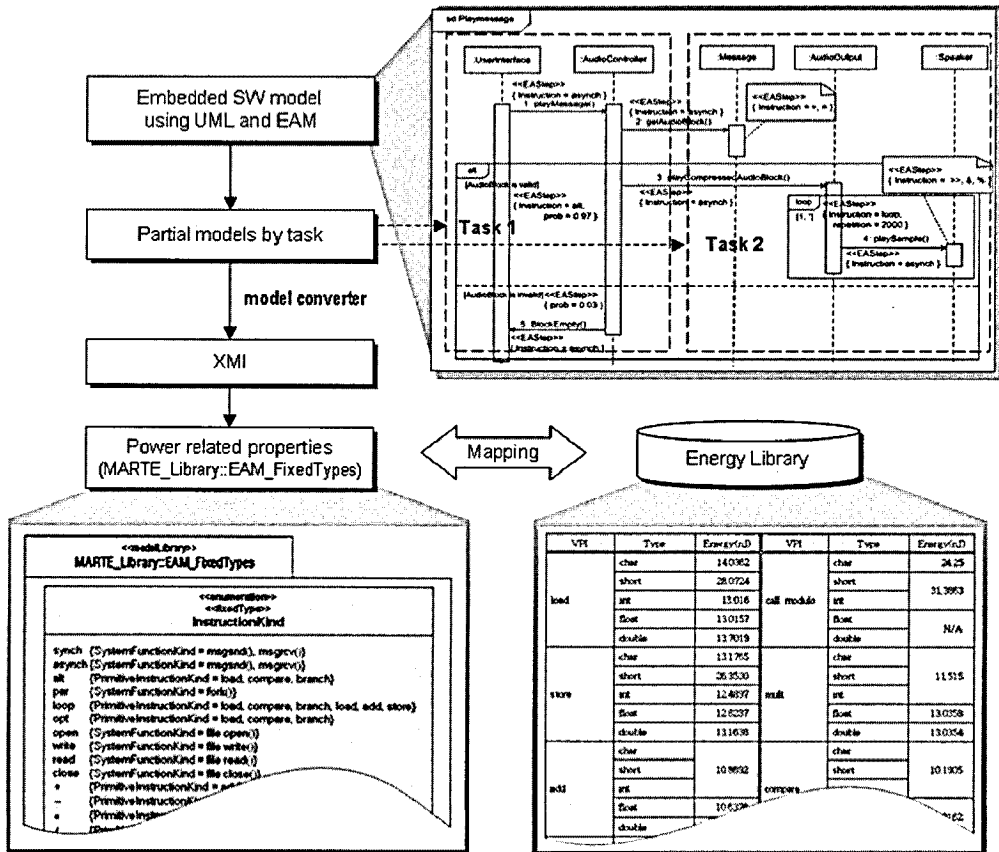


그림 17 MARTE 기반 소모 전력 산출 과정

각의 명령이 수행 시 소모되는 전력량을 포함하고 있다[17].

UML과 EAM 프로파일을 사용하여 모델링된 임베디드 소프트웨어 모델은 그림 16의 절차에 의거하여 태스크들로 분할되고, 각 태스크의 모델들은 모델 컨버터를 통해 XMI 형태로 변환된다. 변환된 XMI에는 모델에 명시한 전력 관련 속성정보가 포함되기 때문에, XMI의 파싱을 통하여 해당 속성 정보를 추출하고, 이 정보들은 그림 17에 나타난 것과 같이 MARTE\_Library::EAM\_FixedTypes를 기반으로 에너지 라이브러리의 대응 요소들과 매핑된다.

그림 17의 과정에 의거한 4.1절의 예제 모델에 대한 명령어 소모 전력은 18.4342 mJ의 값을 얻을 수 있다. 이때 적용된 에너지 라이브러리는 ARM 리눅스 계열의 SA1100 프로세서 기반의 하드웨어 아키텍처를 기준으로 하고 있다. 따라서 4.2절의 식 (1)로부터  $S_i$ 가 0.03이고,  $C_i$ 가 0.05일 때, 예제 모델에 의해 소모되는 전력량  $P(M)$ 은 19.9089 mJ가 된다.

위와 같이 UML 및 EAM 프로파일을 이용하여 설계 모델에 추가(annotation)한 전력 속성 정보를 이용하여 모델 기반의 소모 전력 분석이 가능함을 확인하였다. 추가적으로 수행한 UML 기반 설계 모델의 소모 전력 분석 결과 Tan이 제시한 EMSIM 2.0[18]을 이용한 소스코드 기반의 소모 전력 분석 결과와 비교하여 10%의 오차를 보인다[17]. 비록 소스코드 기반의 소모 전력 분석이 분석 결과의 정확도 측면에서 우수하다고 할지라도, 모델 기반의 소모 전력 분석은 설계 시점에서 에너지 효율적인 설계 모델을 확보하기 위한 용도로 유용하게 사용될 수 있다. 또한 이러한 UML 모델 기반의 소모 전력 분석을 지원하기 위해서는 MARTE 프로파일의 확장을 통한 분석기법이 제공되어야 한다.

## 5. 결론 및 향후 연구

본 논문에서는 UML 모델 기반의 전력 분석을 수행하기 위해 실시간 임베디드 시스템의 비기능적 요구사항을 효율적으로 반영하고 분석할 수 있는 MARTE 프로파일을 확장하였다. 임베디드 소프트웨어의 모델링 시, 모델 요소에 명령어 수준의 패턴을 명세하기 위해 MARTE 프로파일을 확장한 EAM 프로파일을 제안하였고, 추가로 어플리케이션 레벨에서의 태스크가 하드웨어 플랫폼에 할당될 때 모델 요소들에 대한 정보를 유지하기 위해 MARTE 프로파일의 HLAM 패키지를 확장하였다. 이렇게 확장된 결과물은 모델 기반의 전력 분석을 수행할 수 있는 기회를 제공한다. 또한 예제 시스템의 모델링에 적용되어 제안한 방법의 적합성을 확인하였다. 향후에는 분석의 정확성을 향상시키기 위하여 현재 개발된 모델 기반의 소모 전력 분석 도구를 보완

하여 실측값과의 오차를 줄이는 연구를 진행할 것이다.

## 참고 문헌

- [1] H. R. Pyeon, J. P. Kim, and J. E. Hong, "Toward MARTE profile extension for model-based power analysis of embedded software," *proc. of the KCSE 2009*, vol.12, no.2, pp.271-278, Feb. 2009. (in Korean)
- [2] S. Nikolaidis, et. al., "Developing an environment for embedded software energy estimation," *Computer Standards & Interfaces*, vol.28, no.2, pp.150-158, 2005.
- [3] V. Tiwari, et. al., "Power Analysis of Embedded Software: A First Step Towards Software Power Minimization," *IEEE Transaction on VLSI System*, vol.2, pp.437-445, 1994.
- [4] A. Sivasubramaniam, et. al., "Designing Energy-Efficient Software," *Proc. of the IPDPS '02*, pp.115-122, 2002.
- [5] J. Laurent, et. al., "High Level Power Analysis for Embedded DSP Software," *Technical Committee on Computer Architecture Newsletter*, pp.41-46, 2001.
- [6] T. K. Tan, et. al., "Software Architectural Transformations : A New Approach to Low Energy Embedded Software," *Proc. of the DATE '03*, 2003.
- [7] H. Jun, et. al., "Modelling and Analysis of Power Consumption for Component-Based Embedded Software," *Proc. of the EUC Workshops*, pp.795-804, 2006.
- [8] X. Yue, et. al., "OOEM: Object-Oriented Energy Model for Embedded Software Reuse," *Proc. of the IRI '03*, pp.551-558, 2003.
- [9] OMG, "UML Profile for MARTE: Modeling and Analysis of Real-Time Embedded systems," *Doc #ptc-08-06-09*, 2008. available from: <http://www.omg.org>
- [10] OMG, "UML Profile for Schedulability, Performance, and Time," version 1.1, *Doc #05-01-02*, 2005. available from: <http://www.omg.org>
- [11] Taha, S., et. al., "An Open Framework for Detailed Hardware Modeling," *Proc. of the IEEE International Symposium on Industrial Embedded Systems*, Portugal, July 2007.
- [12] S. Bernardi, et. al., "Adding Dependability Analysis Capabilities to the MARTE Profile," *Proc. of the MoDELS '08*, pp.736-750, 2008.
- [13] OMG, "Unified Modeling Language: Superstructure," version 2.1.2 (*formal/ 2007-11-02*) edition, 2007. available from: <http://www.omg.org>
- [14] H. Espinoza, et. al., "Annotating UML Models with Non-Functional Properties for Quantitative Analysis," *LNCS*, vol.3844, pp.79-90, 2006.
- [15] H. I. Rim, A. R. Han, D. H. Bae and J. E. Hong, "Model-based Energy Consumption Estimation

using Instruction-Patterns in Embedded Software Development," *Proc of the KCSE 2009*, vol.11, no.1, pp.122-129, 2009. (in Korean)

[16] I. G. Song, G. Y. Oh, J. E. Hong, and D. H. Bae, "Modeling and Simulation of Platform Specific Model in MPSoC Environment," *Journal of KIISE: Software and Applications*, vol.34, no.8, pp.697-707, 2007. (in Korean)

[17] D. H. Kim and J. E. Hong, "Energy Component Library for Power Consumption Analysis of Embedded Software," *Journal of KIPS*, vol.16-D, no.6, pp.871-880, 2009. (in Korean)

[18] T. K. Tan, A. Raghunathan, et al., "EMSIM: An Energy Simulator Framework for an Embedded Operating System," *Proc. of the International Symposium of Circuits and Systems*, pp.464-467, 2002.



편 호 립

2008년 충북대학교 컴퓨터공학과(학사)  
2008년~현재 충북대학교 컴퓨터과학과 석사과정. 관심분야는 객체지향 모델링, 임베디드 소프트웨어, 저전력 소프트웨어



김 종 필

2006년 충북대학교 컴퓨터공학과(학사)  
2008년 충북대학교 컴퓨터과학과(공학석사). 2008년~현재 충북대학교 컴퓨터과학과 박사과정. 관심분야는 Aspect 기반 컴퓨팅, 임베디드 소프트웨어, 소프트웨어 품질공학, 저전력 소프트웨어



홍 장 의

1988년 충북대학교 전산학과(이학사). 1990년 중앙대학교 컴퓨터공학과(공학석사)  
2001년 한국과학기술원 전산학(공학박사)  
2002년 국방과학연구소 선임연구원. 2002년~2004년 (주)솔루션링크 기술연구소장  
2004년~현재 충북대학교 컴퓨터공학 부교수. 관심분야는 소프트웨어공학, 임베디드 소프트웨어, 소프트웨어 품질공학, 소프트웨어 프로세스 개선, 저전력 소프트웨어