

사
례
발
표

임베디드 어플리케이션을 위한 모델 주도형 개발도구 : 랩소디(Rhapsody)

조 상 윤*

목 차

- 1. 요 약
- 2. 서 론
- 3. Rhapsody의 주요기능
- 4. 결 론

1. 요 약

최근 유비쿼터스, 퍼베이시브 컴퓨팅이란 용어들이 자주 거론되면서 임베디드 시스템이 화두가 되고 있다. 과거에는 임베디드 시스템용 소프트웨어라고 하면 16비트급 이하의 저렴한 프로세서 상에 OS를 사용하지 않고 펌웨어를 작성하는 것을 의미했다. 하지만 고급 프로세서가 점점 저렴한 가격에 출시되고 시스템의 요구사항이 복잡해지면서 근래에는 임베디드 시스템 소프트웨어 작성이 점점 복잡해지고 있다. 이 시대의 임베디드 소프트웨어 엔지니어는 각종 소프트웨어 관련 지식에 하드웨어, OS, 네트워크 지식, 경우에 따라서는 실시간 시스템 이론까지, 실로 많은 것을 알아야만 하는 험겨운 처지에 놓여있다. 이러한 개발 환경을 극복하기 위해 최근 외국과 국내에 많이 알려진 실시간 임베디드 소프트웨어 설계 툴인 Rhapsody의 기능에 대해 알아본다.

2. 서 론

실시간 임베디드 시스템 어플리케이션의 요구

사항들은 놀라운 속도로 증가하고 있다. 짧은 개발 기간 동안 부족한 예산과 자원으로 복잡한 어플리케이션을 만들어 내기란 엔지니어들에게는 실로 많은 부담이 되고 있다. 소프트웨어의 디자인, 분석, 구현(implementation), 테스트, 문서화와 배치에 있어서의 기존방식은 글로벌 경제에서 비교우위를 차지하기 위한 방법으로서 너무 느리며 오류에 빠지기 쉽다.

보다 좋고 빠르면서 적은 예산으로 복잡한 시스템을 개발하는 방법은 시장과 고객의 요구사항을 만족시키는 것이다. 또한 부품수의 폭발적인 증가와 임베디드 어플리케이션의 복잡성은 계속 증가할 것이므로 생산성이 항상 중요한 문제가 될 것이다.

잘 알려진 반도체 회사의 최신 자료에서 반도체 칩의 성능은 매년 70%의 증가를 보이고, EDA툴 사용으로 인한 하드웨어의 디자인 생산성은 매년 30%의 증가를 보인다고 보고하고 있다.

반면, 같은 기간 소프트웨어의 생산성 향상은 10%에도 못 미치는 것으로 보고되고 있으며, 이러한 소프트웨어와 하드웨어의 디자인 생산성의 차이는 칩 디자인과는 다른 지난 30년 간 개선되지

* (주)다한테크 대표이사

많은 소프트웨어의 디자인과 개발 프로세스 때문이다.

과거 소프트웨어/시스템 엔지니어가 사용하는 틀에 의한 약간의 개선이 있었지만 여전히 생산성이 떨어지는 이유는 요구사항 분석, 시스템 설계, 소프트웨어의 수행과 테스트에 연계된 것이 아니라 최하위 레벨에서의 코드 생성, 컴파일, 링크와 디버그에 중점이 두어졌기 때문이며 결과적으로 아직도 시스템 통합과 제품의 특성을 초기에 검증하기에는 어려우므로 매우 중요한 이슈가 되고 있다.

이러한 시대적 요구에 상응화된 제품인 I-Logix(주1)사의 Rhapsody를 이용하여 최적화되고, 도식화가 가능하고, 유지보수와 재사용이 가능하며, 신뢰성 있는 실시간 임베디드 어플리케이션을 개발함으로써 시장대응(Time-to-Market)력을 확실하게 강화, 개선할 수 있을 것이다.

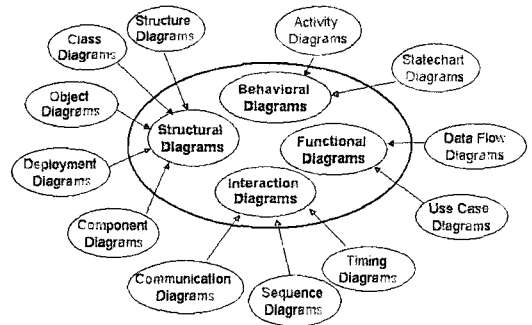
3. Rhapsody의 주요기능

UML 2.0(Unified Modeling Language) 기반 모델 주도형 개발(Model-Driven Development) 환경을 제공하는 Rhapsody는 시스템, 소프트웨어, 하드웨어 그리고 테스트 엔지니어들이 함께 일할 때 최고의 유연성, 최적의 의사소통과 협력관계를 제공하며 동시적 개발 수행이 가능하다.

랩소디(Rhapsody) 틀은 UML을 통한 시스템의 가시적 표현을 제공함으로써 개발 분야에 관계 없이 구현된 시스템에 대해 손쉬운 버튼 동작 환경에서 실행, 구현과 테스트가 가능하므로 모든 시스템 아키텍처를 모델링할 수 있으며, 4가지 언어 즉, C++, C, Java 그리고 Ada를 자동으로 생성할 수 있다. 또한 엔지니어가 요구하는 정확한 환경을 구현하기 위하여 다른 틀과의 인터페이스를 제공하며, 최상의 생산성, 양질의 소프트웨어와 시스템 생산을 위해 증명된 아래 기술된 주요 기능을 제공한다.

3.1 UML 2.0 기반 비주얼 모델링

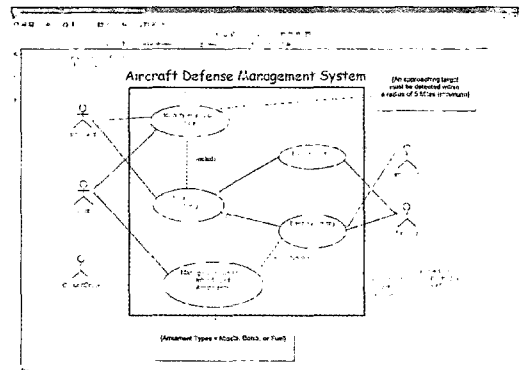
표준 그래픽언어인 UML 2.0을 이용한 시스템 아키텍처 설계, 요구사항 모델링과 분석을 위한 비주얼 환경을 제공하며, 개발자가 직관을 사용하여 시스템을 쉽게 디자인할 수 있도록 편리하고 조합적인 디자인 환경을 제공한다.



(그림 1) UML2.0 Diagrams

(그림 1)은 UML 다이어그램의 뷰(Views)들을 나타낸다.

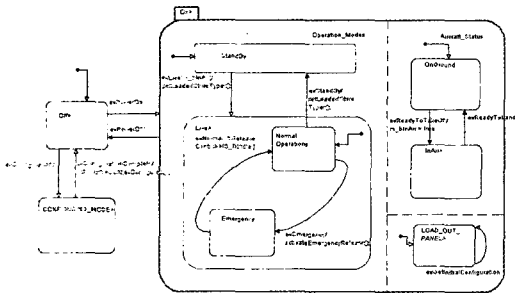
쓰임새도(Use Case)는 기능적 요구사항 분석을 위한 첫 번째 입력 메커니즘이며, 시스템과 소프트웨어 엔지니어들이 개발 시스템에 대한 요구사항을 전체적으로 습득하기 위한 다이어그램이다.



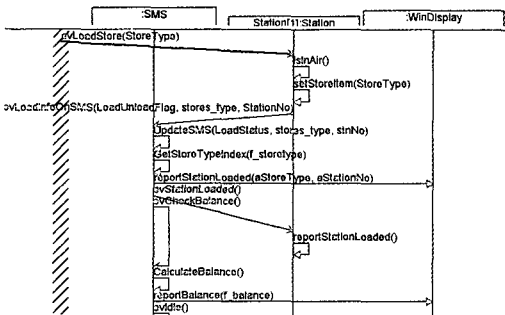
Capture system requirements graphically through Use Case Diagrams, enhancing communications.

(그림 2) ADMS(Aircraft Defense Management System)의 쓰임새도

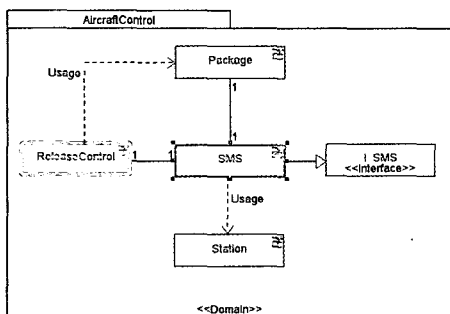
쓰임새도(Use Cases)와 그들에 대응하는 시스템의 동작은 순차도(Sequence Diagram), 협력도(Collaboration Diagram), 활동도(Activity Diagram)와 상태도(Statechart)를 이용하여 상세하게 표현될 수 있다.



[Statechart Diagram]



[Sequence diagram]



[Object Model Diagram]

(그림 3) 시스템 상세 기술을 위한 다이어그램들

시스템 설계를 기술하기 위한 화이트보드 단계에서 이러한 다이어그램들을 사용할 수 있으며, 시스템과 소프트웨어 엔지니어들 간의 아이디어를 개발 프로세스의 모든 단계에서 표현할 수 있다.

프로세싱 노트를 갖는 시스템 아키텍처, 통신 채널, 소프트웨어 컴포넌트들은 사용자가 하드웨어나 소프트웨어를 바꾸는 것을 허락하기 위해 컴포넌트도(Component Diagram)와 배치도(Deployment Diagram)를 구성할 수 있으며, 보다 상세한 아키텍처는 패키지(Structure Diagram), 클래스(Class Diagram) 그리고 객체도(Object Model Diagram)로 기술되어질 수 있다. 더하여 상태도(Statechart), 협력도(Collaboration Diagram), 활동도(Activity diagram), 순차도(Sequence Diagram)를 이용하여 시스템과 소프트웨어의 상세한 동작을 정의할 수 있다.

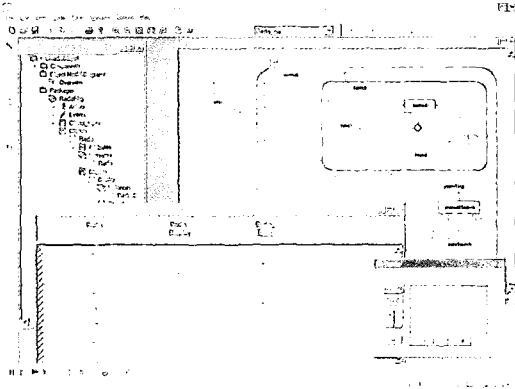
Rhapsody는 어떠한 단계에서든지 정확하고 사용자가 정의한 템플릿(Template)에 따르는 보고서를 자동으로 생성해 주며, 사용자가 작성한 코드이든 자동으로 생성된 코드이든 UML 모델과 연동할 수 있다. 따라서 Code도 한번에 마우스 클릭으로 볼 수 있는 또 다른 뷰(View)며 더하여 모든 개발단계에서 사용할 수 있는 강력한 테스트 성능을 제공한다.

3.2 실행 가능한 시스템 모델 설계

UML 모델링만으로는 시스템을 충분히 설명하기에는 부족하므로 설계된 모델이 실행 가능하다면 이 문제는 시스템의 이해뿐만 아니라 검증까지도 가능할 것이다.

소프트웨어 생산성의 실제적 개선을 위해서는 개발단계 초기에 오류가 제거되어야 하는데 UML 모델의 실행을 통하여 이것이 가능하게 할 수 있다. 모델의 실행은 시스템, 소프트웨어 그리고 테스트 엔지니어가 개발 프로세스의 어떠한 단계에서도

시스템을 검증할 수 있게 한다. 또한 시스템 구성을 디자인 하기위해 사용하는 그래픽 언어(UML)를 사용하여 어플리케이션을 디버깅할 수 있다.



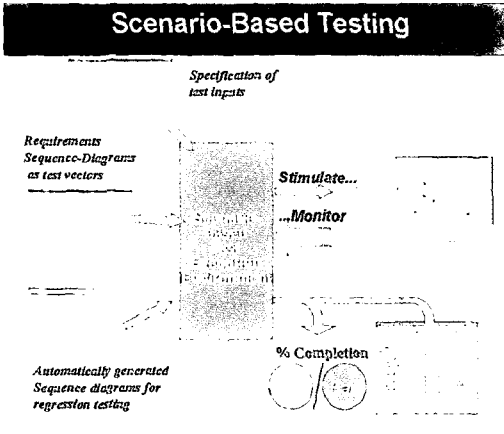
(그림 4) Statechart와 Sequential Diagram 그리고 GUI를 이용한 모델의 실행과 검증.

들어 내는 것을 가능하게 한다. 더하여, 소프트웨어 엔지니어는 시스템의 검증을 위해 테스트로 단계로 보내지기 전에 컴포넌트의 구조와 상호 인터페이스, 그리고 이들의 행위를 검증하고 실행할 수 있다. 이 시점에서 테스트 그룹은 블랙박스 레벨에서 (그림 5)에서와 같이 시나리오 실행에 의해 시스템을 그래픽하게 검증할 수 있다.

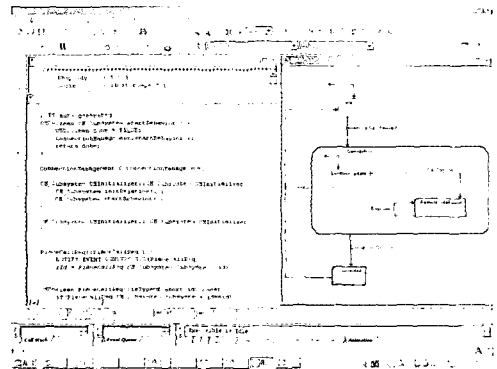
3.3 다이내믹 모델, 코드조합 (Roundtrip Engineering)

소스코드 레벨이든 UML디자인 레벨이든 상관없이 개발환경에 대한 자율성을 보장한다. 그러므로 핸드코딩으로 상당수의 일을 하더라도 개발 기간에는 큰 문제가 없다.

다이내믹 모델, 코드 조합은 Rhapsody가 유일하며, 이 기술을 이용하는 코딩은 디자인 프로세스의 back-end뿐만 아니라 전 과정에 걸쳐 다이내믹하게 짜여진다.



(그림 5) Rhapsody에서의 시나리오 기반 테스트 환경



(그림 6) 다이내믹 모델, 코드조합

Rhapsody는 UML로 디자인된 시스템 설계물을 검증 가능한 실행 모델로 바꾸어 준다. 예를 들어, 모델 실행을 하는 동안 (그림 4)와 같이 상태도는 현재 동작되는 상태를 하이라이트로 표시하고, 시퀀스 다이어그램에서 메시지는 컴포넌트에서 다른 컴포넌트로 이동되며, 인스턴스의 생성과 소멸, 속성 값의 변화 등을 확인할 수 있다. 이것은 시스템 엔지니어가 그래픽하고 실행 가능한 명세서를 만

모델과 코드는 Rhapsody 개발환경에서 보면 단순히 같은 디자인의 두개의 뷰(View)지만 모델을 바꾸면 코드가 업데이트 되고, 코드를 바꾸면 모델이 업데이트되는 다이내믹 구조를 가지고 있고 항상 코드자체 또는 코드생성 파라미터를 바꾸기 위해 전체 코드를 제어한다. 이러한 양방향 반복접근 방법은 디자인과 코드가 항상 동시성을 가지고 개

발됨을 보장함으로써 보다 정확한 코드생성과 UML 모델을 생성할 수 있게 한다.

전통적 개발 프로세스 또는 블랙박스 코드 생성 기술과 같은 가상적 소프트웨어 개발 시스템은 개발자가 원하는 코드를 생산하기 위해 시험과 오류를 반복해야만 한다. 이 전통적 프로세스는 요구되는 코드의 출력을 위해 지속적인 모델수정을 끊임 없이 반복해야 하므로 디자인된 모델이 실제 코드와 매칭이 되지 않는 경우도 발생하게 된다.

Rhapsody는 그래픽 레벨이든 코드 레벨이든 상관없이 (그림 6)과 같이 양방향 뷰(View)로 빠르게 수정할 수 있으므로 디자인과 코드를 일치성을 가지고 반복 개발을 함으로써 설계 단계의 오류를 최소화 할 수 있다.

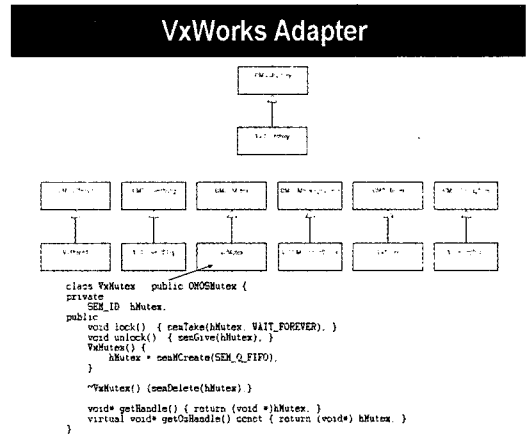
3.4 어플리케이션의 자동통합

임베디드 시스템을 개발하다보면 많은 상용 RTOS(Real-Time Operating System)을 접하게 된다. 이러한 상용 RTOS의 구조를 모두 이해하고 시스템을 개발하려면 많은 인력과 개발비를 투입해야 할 것이다. 짧은 개발시간, 오류에 대한 부담 등 개발 시에 일반적으로 부딪치는 문제들에 대해, Rhapsody는 양질의 코드 자동 생성과 보고서 자동 생성 등의 많은 자동기능을 제공한다. 구체적으로 소프트웨어 어플리케이션의 설계, 런타임 라이브러리, makefiles, 통신 기초구조, 코드구현과 디자인 도큐먼트 등을 자동으로 생성하며, 동시에 보다 구체적인 어플리케이션을 수행하기 위한 핸드코드의 삽입도 가능하다.

Rhapsody는 어떠한 임베디드 타겟에 대해서도 자동적인 어플리케이션을 생성을 할 수 있으며, 실시간 프레임워크(Real-Time Framework)이 이것을 가능하게 해준다. 이 기술은 실제 타겟의 동작환경에서 운용되는 어플리케이션의 개발 기간을 최단 시간으로 줄여주며, 서로 다른 물리적 아키텍처를

같은 로직 디자인으로 전개할 수 있다.

전통적으로, 대부분의 개발자는 개발시간의 상당부분을 어플리케이션 구조 외의 컴포넌트를 만드는데 시간을 허비한다. 이것은 분산형구조, 컨테이너 수행, 디자인 패턴, OS 인터랙션, 상태정의, 시간운용과 프로세스 간의 통신 등의 메시지 구조를 포함한다.



(그림 7) VxWorks Adapter

제공되는 리얼타임 프레임워크는 많은 상업용 RTOS(Real-Time Operating Systems)에 대하여 소스코드와 컴파일된 형태로 지원한다. 이 프레임워크는 전체 프로젝트에 부합되어 사용되어야 할 수 있도록 되어 있으며, 코드생성 프로세스의 부분과 임베디드 실시간 시스템에 대하여 특별하게 디자인되어 있다. 또한 Rhapsody 리얼타임 프레임워크는 실제적 소유권이 없고 상용 RTOS에 부합하기 위하여 확장하거나 재구성할 수 있으며, RTOS를 사용하지 않은 Target 시스템에 대해서도 사용할 수 있다. (그림 7)은 제공되는 Adapter 구조의 한 예이고 <표 1>과 <표 2>는 각각 제공되는 상용RTOS Adapter와 IDE(Integrated Development Enterprise) 목록이다.

<표 1>은 제공되는 상용 RTOS Adapter목록

제조사	제품
ARC	MQX
Accelerated Technology	Nucleus Plus
CMX System	CMX RTX
Enea(OSE Systems)	OSE Delta
Green Hills/ExpressLogic	Integrity, ThreadX
Lynux Works	Lynx OS
MontaVista	MontaVista Linux
QNX	Neutrino
Red Hat	Red Hat Linux
Segger	EmbOS
Sun	Solaris (RTX and ETS)
WindRiver	VxWorks, pSOS(pPISM)
ITRON	Microltron
Microsoft	Windows CE.NET Windows NT 4.0/2000/XP

<표 2> 호환되는 IDE(Integrated Development Enterprise)

Tools

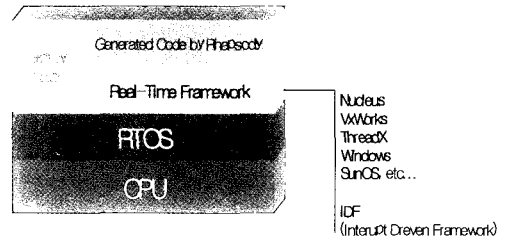
제조사	제품
Altium	Tasking EDE
Green Hills	Multi 2000
Keil	µVision
WindRiver	Prism-Tornado

3.5 디자인 레벨 디버깅과 테스트

실행 가능한 모델은 시스템 디자인 초기에 그래픽 디버깅을 가능하게 함으로써 양질의 디자인을 수행하고 개발 사이클을 현저하게 줄여준다. 더하여, 타겟과 개발 플랫폼(HOST)모두에서 디자인 레벨 디버깅을 가능하게 한다.

호스트 환경에서 만족하는 디자인의 그래픽 디버깅을 실행함으로써 어플리케이션을 구성하는데 있어 획기적인 수정이 가능하다. 실행 가능한 모델은 어플리케이션을 하드웨어도 만들기 전에 시스템의 검증과 디버깅을 가능하게 함으로써 하드웨어가 완성되면 선택된 타겟 환경에서 같은 실행 가능한 모델을 생성하는 것은 두말할 것도 없다.

Rhapsody는 IDE(Integrated Development Enterprise)를 통하여 소스레벨 디버깅과 혼합하여 디자인할 수 있으며, 혼합 디버깅을 이용하여 보다 복잡한 어플리케이션 개발을 위한 다이내믹 코드와 시스템 모델링을 가능케 한다. (그림 8)은 제공되는 RTOS Adapter와 CPU에 올라갈 프로그램과의 관계도다.



(그림 8) 최적화된 RTOS Adapter(Real-Time Framework)

4. 결 론

UML 2.0을 기반으로 하는 Rhapsody는 실시간 임베디드 어플리케이션 시스템과 소프트웨어를 설계하고 생성하기 위한 솔루션이다. 최근의 유비쿼터스의 일환으로 임베디드 시스템의 컴포넌트화(Component Based Development)와 모델화(Model-Driven Development)하는 개발 방법은 이미 우리나라에서도 조기에 도입된 상태며 시스템 개발 초기에 사용되고 있다. 그러나 아직도 많은 엔지니어들이 이를 적절히 활용하고 있지 못하고 있는 현실이다.

Rhapsody는 이미 해외에서 검증된 툴로써 자동차, 항공기, 원자력, 국방, 핸드셋 등의 분야에서 개발표준 툴로 널리 사용되고 있다.

최근에는 개발초기에 임베디드 시스템의 설계뿐만 아니라 시스템의 전체적인 테스트 및 검증을 위한 프로그램이 출시되고 있다. I-Logix사는 Rhapsody ATG(Automatic Test Generation)과 TestCuductor를 출시한 상태며, 이미 안전필수(Safety-Critical)분야에 활용되고 있다.

참고문헌

- [1] Douglass, Bruce Powel, Real-Time UML 3rd Edition: Advances in UML for Real-Time Systems, Addison-Wesley, 2004
- [2] Douglass, Bruce Powel, Doing Hard Time: Developing Real-Time Systems with UML, Objects, Frameworks, and Patterns, Addison-Wesley, 1999
- [3] Douglass, Bruce Powel Real-Time Design Patterns: Scalable Architectures for Real-Time Systems Addison-Wesley, 2002
- [4] D. Harel and Amnon Naamad The STATE MATE Semantics of Statechart, ACM Trans. soft. eng. Method., 1995.
- [5] QING LI의 RTOS를 이용한 실시간 임베디드 시스템 디자인 에이콘출판(주)(내서), 2004년 2월 13일
- [6] <http://www.ilogix.com>

저자약력



조 상 운

1992년 인하대학교 컴퓨터공학과졸
현 (주)다한테크 대표이사

(주1) I-Logix사는 Object Management GroupTM(OMGTM), the Bluetooth SIG의 회원사이고 Embedded Linux Consortium 의 공동 설립자이며, UMLTM 공동 창시회사다.