

UI 기반의 임베디드 시스템 가상 프로토타이핑¹⁾

이영란*, 이정배*, 엄기철*, 김상복*

*선문대학교 컴퓨터정보학부

{yrlee, jblee}@sunmoon.ac.kr, sbkim@anybill.com

A Virtual Prototyping of Embedded System based on User-Interfaces

Young-Ran Lee*, Jeoung-Bae Lee*, Ky-Chul Eum*, Sang-Bok Kim*

*Sunmoon University, Division of Information & Computer Science

{yrlee, jblee}@sunmoon.ac.kr, sbkim@anybill.com

본 논문에서는 임베디드 시스템의 신뢰성 있는 개발을 위한 방법으로 가상프로토타이핑을 제안한다. 가상프로토타이핑으로 컴포넌트의 재사용을 통한 IT 정보의 활용도를 높이고 복잡한 시스템의 유지보수 비용 및 개발 비용을 절감하기 위하여 User-Interface 기반의 개발방법을 사용한다.

1. 서론

IT기술은 다양한 사회와 경제적 변화에 따라 다양한 욕구를 충족시키기 위해 복잡화 및 광대역화 되고 있으며, IT 산업은 임베디드 시스템을 중심으로 이러한 추세를 통합하고 있다. 특히 임베디드 소프트웨어 산업의 육성은 2010년 국민소득 2만불 달성이 라는 국가 발전 목표의 실현을 위한 핵심 산업으로 2003년 8월 10개 신성장동력 산업의 핵심육성 분야로 선정되었다.

저비용으로 고부가가치의 제품 생산을 위한 임베디드 소프트웨어 개발시, 하드웨어와의 동시 설계 및 역할 분담이 필수적이며, 요구분석, 하드웨어와 소프트웨어의 역할 및 동시설계와 재통합의 과정에 따른 검증과 디버깅 과정이 개발방법론에 포함되어야 하나 개발 환경에 따라 다양한 방법론이 제시 될 수 있으며, 개발자의 정확한 이해와 기술 변화에 대한 인식이 요구된다.

본 논문은 현재 디지털 기술의 핵심인 임베디드 시스템의 특성을 반영한 가상프로토타이핑의 특징과 필요성을 알아보고 3장은 임베디드 시스템의 컴포넌트 재사용의 특성을 중심의 개발 방법으로 UI 중심의 개발과 플랫폼 기반의 개발 방법의 특성을 살펴

보고, 4장에서는 소프트웨어의 동시 설계와 동시 시뮬레이션 및 디버깅이 가능한 실제 시스템을 구현과 테스트를 위하여 세탁기를 UI기반의 개발 도구를 이용하여 구현하고 테스트한다. 5장에서는 UI 기반개발 방법이 신속한 개발과 저비용의 개발을 가능하게 하는 편리함에도 불구하고 실제 개발현장에서 거의 사용되지 않는 현실을 보고 추후 과제에 대하여 논한다.

2. 가상프로토타이핑(Virtual Prototyping)

프로토타입(prototype)이란 제품 개발과정중 제품 개발이 완료되고 양산단계 전에 제작되어진 축소형 또는 실물크기의 모델로, 제품의 외형 및 기능 시험을 위한 테스트와 소비자의 반응을 조사하기 위한 모델을 제작하는 것으로 제품 생산과정의 일부로 볼 수있다. 프로토타이핑을 제품의 각 기능을 테스트와 검증 및 설계상 오류 수정을 컴퓨터를 이용하여 데이터 수정 작업으로 수행하는 것을 가상 프로토타이핑이라고 한다.

가상프로토타이핑은 제품 개발비용의 80%를 차지하는 초기 비용과 시간과 소요인력을 최소화함으로써 시장에서의 제품 경쟁력을 향상하는데 목적이 있다. 특히 임베디드 시스템의 복잡도가 증가하고, 추가되는 기능의 폐기 및 생성 주기가 짧은 현재 시장에서 수정이나 업그레이드가 빈번히 발생함으로써

본 연구는 정보통신부 및 정보통신연구진흥원의 대학 IT 연구센터 지원사업의 연구 결과로 수행되었음

소프트웨어의 유지 보수의 유연함을 위하여 재사용 가능한 컴포넌트화 중심의 가상프로토타이핑 기술은 제품 개발과정의 필수요소라 할 수 있다.

2.1 임베디드 시스템 가상프로토타이핑 환경

임베디드 시스템은 크기, 메모리, CPU, 네트워크 대역폭, 저전력, 실시간 등 여러 제약사항을 고려한 선택이 되거나 개발이 진행되어야 하며 이러한 제약사항을 고려한 자원 모델 기반의 방법 및 하드웨어 소프트웨어의 co-design과 co-simulation의 방법에 관한 연구가 활발히 진행되고 있다.

3. UI기반의 가상프로토타이핑

3.1 UI(User Interfaces)개요

현대사회는 컴퓨터가 직접 또는 간접적으로 생활과 긴밀한 관계를 가지면서 컴퓨터와 인간의 상호작용이 관심을 끌면서, 이를 위한 연구의 한 영역으로 인간-컴퓨터 상호작용(HCI:Human-Computer Interaction)가 대두되었다. HCI 시스템은 독립된 두 개체를 하나의 통합된 시스템으로 구축하기 위한 인터페이스의 필요성이 중요한 개념이 되었다. 따라서 HCI기반의 개발시 적절한 인터페이스로 UI(User-Interfaces) 개념이 나타났으며, UI는 인간과 시스템 간의 공통점과 사용자와 각각의 시스템 사이의 정보채널로 표현되어 사용하기 편리한 시스템을 만들기 위하여 사용자의 인지적 측면에서 디자인하고 편리성을 평가하는 개념으로 정의된다.

이러한 UI 중심 설계방법은 제품 제작의 전 과정에서 확대되고 있으나 UI 디자인에 이용될 적절한 도구 및 시제품 생산을 위한 인터페이스를 위한 도구의 부재가 문제점으로 나타나고 있다.

3.2 UI 특징

UI는 객체 지향적 개발방법의 발전된 형태로 CBD(Component-Based Development)기반의 개발 방법이다. UI 기반의 개발방법은 시스템의 변경이 잦은 임베디드 시스템의 기능의 첨가 및 수정이 간단하며, 재사용을 목적으로 컴포넌트화 되어 IT자산의 활용도를 높이고, 시스템의 유지 보수비용의 최소화로 경제적 효과를 극대화 한다. 많은 UI기반 틀이 시뮬레이터와 디버거, 프로젝트 관리자, 컴파일 기능을 포함하고 소스내비게이션 기능을 통합함으로써 다양한 플랫폼을 지원하고 전문적인 프로그래밍 언어에 능숙하지 않아도 쉽게 시스템을 구현할 수 있

다.

UI기술과 VP기술을 통합한 개발 방법은 실질적인 문제 해결과 강력한 시뮬레이션을 제공하여 제품의 초기 개발비용을 최소화 하는 효과가 있다. 또한 하드웨어와 소프트웨어 동시 시뮬레이터 및 디버거가 통합되어 제공됨으로써 기존의 개발 방식보다 편리한 개발 모형을 제공한다. 이러한 통합 개발 환경은 사용자 혹은 개발 의뢰자가 현실감있는 제품과 환경에서 최적화된 플랫폼을 가상의 기반으로 구현 가능한 멀티플랫폼 지원과 개발환경을 제공하고, 개발과정에서 생성된 정보는 실제 시스템 구현시 참조가 가능한 문서로 저장되어 제공된다.

현재 Rapid Plus, Rhapsody 등의 상용화된 가상 프로토타이핑 도구가 있으며, 이를 이용하여 임베디드 시스템 개발의 생산성을 향상시킬 수 있다.

UI 기반의 개발 환경을 제공하는 도구의 특징은 다음과 같다.

- GUI 기반의 point-and-click 환경으로 요구되는 컴포넌트와 동작을 생성
- State Chart를 이용한 동작의 설계와 분석 및 문서화 가능
- 하드웨어와 소프트웨어의 동시 시뮬레이션이 가능하며 개발 초기부터 오류수정과 실시간 동작 테스트를 통하여 최종제품의 강력하고 완벽한 시뮬레이션이 가능
- 외부 오브젝트를 위한 인터페이스 코드의 최적화
- 다양한 플랫폼과 실시간 OS가 제공됨으로써 목적에 맞는 시스템 구현을 위한 환경 제공
- 소스레벨 디버거 기능 및 3th-party 디버거 지원

임베디드 시스템 개발에서 UI 중심의 개발방법은 기존의 범용 프로그래밍 틀인 Visual Basic, Visual C++, Delphi, Java 등의 개발도구를 이용 개발하는 방법보다 새로운 모델 구현이나 다른 모델의 프로토타이핑시 편리함을 제공한다.

본 논문에서는 임베디드 가상프로토타이핑 개발 도구인 e-sim사의 Rapid Plus로 컴포넌트의 재사용의 유연함과 확장 가능성을 지원하는 UI 기반의 세탁기 가상프로토타이핑 모델을 구현한다.

4. 세탁기 프로토타이핑 구현

UI기반의 개발도구인 Rapid Plus를 이용하여 실제 사용가능한 세탁기 애플리케이션의 프로토타이핑 시스템을 설계한다.

4.1 애플리케이션 개발과정

Rapid Plus는 애플리케이션을 제작하기 위한 6가지 기본개념을 제공한다.

- Object : 어플리케이션의 UI를 위한 가시적 타입 표현
- Modes : 어플리케이션의 동작가능 상태정의를 위한 계층적 형태
- Transitions : 어플리케이션 모드 사이의 transitions(전이) 생성
- Tirggers : 전이가 발생할 때의 트리거 정의
- Activities : mode의 동작이 만들어졌을때 발생
- Runtime Test : 프로토타입 내부의 구현 어플리케이션 테스트

이와 같은 6가지 개념을 바탕으로 RP(Rapid Plus)는 프로토타이핑을 모델링화하고 각 모델링에 적합한 동작 설계를 위한 편리한 환경을 제공한다.

4.2 세탁기의 가상 프로토타이핑 설계

세탁기의 외관 디자인 및 전원, 세탁코스, 세탁시간, 예약세탁, 탈수 등의 일반적인 디지털 가전의 세탁기 메뉴를 구성하여 실제 세탁수행 테스트 시스템을 구현한다.

4.2.1 구현환경

- 메모리 : 512MB
- 운영체제 : Windows XP
- 개발도구 : Rapid Plus 8.0

4.2.2 세탁기의 정적 모델링

Layout Manager의 Object를 이용한 각 제품의 외형을 설계하고, Object의 상속관계를 나타내는 mode tree를 구성한다.

<그림 1>은 Layout Manager를 이용하여 세탁기의 외형을 모델링하는 과정이다.



<그림 1> 세탁기 정적 모델링

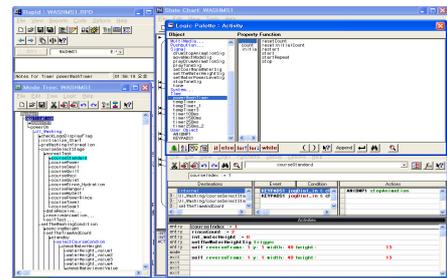
<그림 1>은 세탁조에 물이 채워지고, 세탁이 시작되는 동작을 구현하기 위하여 세탁조와 세탁기 제어를

위한 제어부를 설계한다.

가시적 타입을 위한 그래픽 Object는 그래픽 라이브러리에 등록되어 있으며, 사용자가 필요로 하는 Object가 없을 경우 직접 만들어 라이브러리에 등록하여 사용할 수 있는 UDO(User-Object Library)를 제공한다.

4.2.3 동적 모델링

<그림 2>는 세탁기 어플리케이션의 동작의 상세 구현을 보여준다. tree의 mode를 선택하고 logic editor를 통하여 transition, trigger, event, activity를 Logic Pallet에서 선택하여 구현하고, Prototyper로 테스트 후 오류 및 동작 수정이 가능하다.



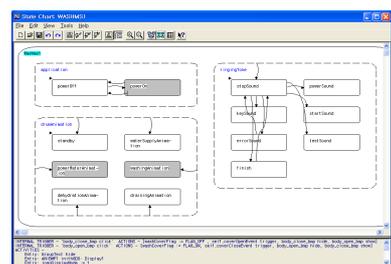
<그림 2> 기능과 동작의 상세 구현

구현을 위한 function과 property는 API에 등록이 되어 있으며, UDO와 같이 시스템이 요구하는 함수가 API에 없을 경우 사용자가 제작하여 API에 등록하여 사용할 수 있는 UDF(User-Define Function)를 제공한다.

<그림 2>의 트리구조에서는 최상위 객체인 Washms1로부터 각 UI와 동작들이 상속되는 구조를 볼 수 있다. 모든 동작은 상속 관계에 있으며, 복잡한 설계 과정이 계층적으로 구성되어 mode tree를 통하여 전체 시스템의 구조를 볼 수 있다. <그림 2>의 트리에서 세탁기 어플리케이션의 최상위가 powerOff/powerOn 상태로 부터 파생되는 것과 같이 동시에 실행이 불가능한 상반된 역할의 같은 계층에서 화살표로 표시하며 Exclusive mode임을 표시한다.

4.2.4 구현 시스템의 모델링

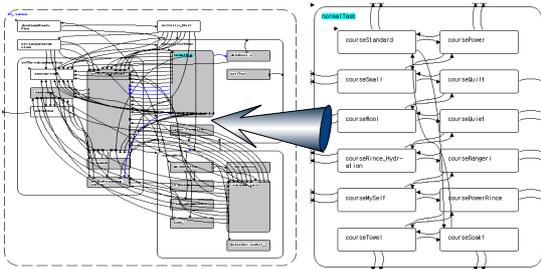
<그림 3>은 구현된 세탁기 프로토타이핑 시스템의 전체를 모델링하였다.



<그림 3> 세탁기 전체의 모델링

Washms1에 각각 application drumAnimation, ringingTone으로 구성된 3개의 mode가 있다. application은 세탁기의 상세 설계 동작을 구현하고 있으며, drumAnimation은 세탁조의 물이 채워지는 상태와 세탁이 실행되고 멈추는 상태를 표현한다. ringingTone은 세탁기 실행 동작 상태를 음성안내 동작을 구현하고 있다.

<그림 4>의 좌측은 세탁기 application의 전체 동작 설계를 보여주고 있으며, 우측은 현재 선택된 모드인 전체 시스템의 일부인 courseStandard mode 내의 Exclude mode인 nomalTask의 정형명세이다.



<그림 4> 세탁기의 상세모델링

4.2.5 세탁기 동작실험

<그림 5>는 세탁기 어플리케이션의 실행 과정 중 세탁 준비단계를 보여주고 있다. 세탁기가 현재 동작하지 않는 상태에서 수온을 선택하고 물높이를 '중', 물살을 '중'으로 선택하여 코스 선택후 세탁을 실행하면 세탁조에 Logic editor를 통하여 설정된 적정량의 수위가 되면, 세탁기 세탁이 실행된다.



<그림 5> 세탁기 동작실험

5. 결론

고난도의 기술과 지식을 요구하는 임베디드 시스템의 개발은 신기술을 탑재한 제품개발이 신속히 이루어져야한다.

따라서, 개발기간과 비용을 절감하는 방법으로 가상프로토타이핑 방법론은 필수적이며, 공학적 관점에서 가상프로토타이핑을 통합한 설계 방법으로, 하드웨어와 소프트웨어의 동시 시뮬레이션 및 디버깅이 가능한 방법을 제안하였다. 복잡한 시스템의 빠르고

쉬운 개발을 위한 UI 기반도구의 특징을 알아보고, e-sim 사의 Rapid Plus로 세탁기의 가상프로토타이핑을 구현하였다.

향후 하드웨어 소프트웨어의 동시 시뮬레이션을 초월하는 가상과 실물프로토타이핑의 동시 시뮬레이션을 위한 인터페이스 설계 및 구현을 위한 지속적인 연구로 제품의 시장경쟁력 뿐아니라 블루오션(blue ocean) 영역인 임베디드 소프트웨어시장 선점과 표준화 및 전 산업 분야로의 적용확대로 임베디드 소프트웨어 산업의 국제적 경쟁력 강화로 국가 경제발전의 원동력이 될 것이다.

[참고문헌]

[1] 이정배 등, “국내 임베디드 S/W산업 실태조사에 관한 연구”, 한국소프트웨어진흥원, 2004.3.
 [2] 2002 Embedded Software Tools Worldwide Forecast, Gartner Dataquest Market Statistics 110850, 2002. 10.
 [3] Worldwide Embedded Software Tools Outlook, 2002, Gartner Dataquest Alert, 2002. 10.
 [4] The 2000 Embedded Software Strategic Market Intelligence Program : Embedded Operating System, Software Development Tools, and Desing Automation Tools, VDC, 2000.
 [5] McGibbon, B., “Status of CBSE in Europe”, Component-Based Software Engineering, Addison-Wesley, 2000.
 [6] 고정훈 등, “승용차의 가상 프로토타이핑”, 한국자동차공학회논문집 제7권 제5호, pp.230~239
 [7] Brown, A., Large-Scale, Component-Development, Prentice Hall, 2000.
 [8] <http://www.e-sim.com>
 [9] <http://www.livecomus.com>
 [10] <http://www.embeddedworld.co.kr>
 [11] 원종혁 등, “사용자 인터페이스 향상을 위한 3차원 VP 시뮬레이터 설계”, HCI2000 학술대회
 [12] 이희웅 등, “Virtual Prototyping이란?”, 전자공학회지 제25권 제2호, pp191~199, 1998.2.
 [13] David Harel, “Statecharts : A Visual Formalism for Complex System”, Science of Computer Programming, 1987.8.
 [14] 김동건 등, “제품 디자인을 위한 개체지향적 Virtual Prototyping에 관한 연구”, 한국과학기술원, 1996.
 [15] F.Jahanian and A.K.Mok, “Modechart: A Specification Language form Real-Time Systems”, IEEE Transactions on Software Engineering, Nov.1989.