

가상 프로토타입핑을 위한 Statechart 도구

김 남 규 °, 박 홍 진, 김 영 찬
중앙대학교 컴퓨터공학과

The Statechart Tool for Virtual Prototyping

Nam-Gyu Kim °, Hong-Jin Park, Young-Chan Kim
Dept. of Computer Science and Engineering, Chung-Ang Univ.

요 약

실제품과 동일한 물리적 특성과 기능적 특성을 갖는 디지털 모델을 만드는 가상 프로토타입핑 기술은 내장형 시스템이 복잡해질수록 더욱 중요시되는 기술이다. 가상 프로토타입핑 요소기술중 정형 명세언어인 Statechart는 제품의 기능적 특성을 나타낸다. 그러나 기존 가상 프로토타입핑에서 사용되고 있는 Statechart에서의 재사용은 단지 복사하여 붙이는 방법만 있어서 재사용시 시스템 설계자의 세밀한 분석이 요구된다.

본 논문에서 구현한 가상 프로토타입핑을 위한 Statechart 도구는 제품의 생명주기를 줄이고, 제품의 시장적기진입을 위해 컴포넌트 기법을 이용하여 보다 빠른 시스템의 설계를 지원한다.

1. 서론

가전 제품을 비롯하여 통신 및 멀티미디어 시스템 내부는 마이크로 프로세서, 주문형 반도체 등으로 이루어진 하드웨어 및 응용 프로그램과 사용자 인터페이스 코드로 형성된 소프트웨어가 어우러져 복합적으로 구성된 내장형 시스템(embedded system)으로 이루어진다. 내장형 시스템을 이용한 제품의 생명주기(life cycle)는 매우 짧아 신제품 출시 속도는 30%이상 빨라지고, 복잡도도 매3년마다 2배정도 증가하고 있다[1]. 시장적기진입(time-to-market)이 무엇보다도 중요해 지고 있는 요즘 제품을 실제로 만들기 전에 미리 컴퓨터를 통해 제품을 평가할 수 있는 가상 프로토타입핑은 제품의 경쟁력뿐만 아니라 소비자의 만족도를 향상시킬 수 있는 기술이다.

가상 프로토타입핑이란 실제품과 물리적 특성(외관, 소리 등)과 가능한 동일하며, 실제품과 동일한 기능적 특성을 갖는 시뮬레이션 디지털모델이다.

가상 프로토타입핑의 중요 요소기술로는 제품의 물리적 특성을 나타내는 가시화 기술과 제품의 기능적 특성을 나타낼 수 있는 정형 명세언어 기술, 위의 두가지 요소기술로 제품에 내장될 프로그램을 만드는 자동코드 생성기술등이다.

본 논문은 가상 프로토타입핑의 기술 중 정형 명세언어에 관한 것이다. 가상 프로토타입핑에서의 정형 명세언어기술은 Statechart를 이용하고 있다[2]. David Harel이 제안한 Statechart는 이벤트에 대해 동작을 명세하는 언어로서 계층적 상대도를 제공함으로 복잡한 제품에 대해 보다 분명하고 효과적인 방법을 제공할 수 있다. Statechart는 국제 표준화 기구인 OMG(Object Management Group)에서 객체지향 모델링 언어(UML)의 일부분으로 포함되어 사용하고 있다[2][3][4].

그러나 기존 가상 프로토타입핑에서 사용되고 있는 Statechart는 작성된 정형명세를 재사용하기 위해 복사해서 붙이는 방법(Copy & Paste)을 이용하고 있다. 이러한 방법은 개발자가 기존 복잡한 정형명세 설계에 대한 세밀한 분석이 필요할 뿐만 아니라, 새롭게 작성될 시스

본 연구는 한국과학재단 특정기초연구(1999-2-303-008-3)지원으로 수행되었음.

템 설계에 적용하기도 어렵기 때문에 불필요한 투자와 시간이 소요될 수 있다.

본 논문은 이미 작성된 Statechart를 재사용 하기 위한 방안으로 컴포넌트 기법을 사용하여 보다 빠른 시스템의 설계를 지원하는 도구를 구현한다.

본 논문의 구성은 다음과 같다. 2장에서는 관련 연구로 기존 도구를 기술하며 3장에서는 도구의 전체적인 구성과 예제를 통해 구현된 도구를 설명한다. 마지막으로 4장에서 결론을 맺는다.

2. 관련 연구

Rapid PLUS은 이스라엘 회사인 Emultek에서 개발한 가상 프로토타입핑 도구이다[5]. Statechart를 기반인 Rapid PLUS는 휴대폰, 전자 시계, 항공기의 제어부분 등 다양한 내장형 시스템을 실제 제품이 나오기 전에 미리 테스트와 디버깅을 할 수 있다. 또한, 기존에 많은 시간과 노력이 필요한 제품의 문서화 작업을 자동화시킴으로써 시스템 개발자는 보다 쉽게 사용자 설명서 등을 제작할 수 있다. Rapid PLUS에서는 시스템의 기능을 설계하기 위해 모드 트리(mode tree)를 통한 Statechart를 생성시킨다. 따라서, 이 도구에서는 Statechart를 변경하기 위해서는 Statechart 자체를 변경할수 없고, 모드 트리를 변경을 통해 이루어지며 Statechart를 재사용하기 위해서는 모드 트리를 복사하여 사용한다.

MagicDraw는 Statechart가 포함된 UML 모델링을 지원하는 CASE도구이다[6]. MagicDraw는 C++, Java, CORBA IDL로 프로그램을 생성시킬 수가 있으며, UML 도구 중 가장 널리 사용되는 Rational Rose와 호환성을 지니고 있어서 확장성이 뛰어나다. 실시간 시스템, 클라이언트/서버 및 분산 어플리케이션 설계시 유용한 도구이다. 그 외 ObjectDomain, Xjcharts등이 있다[7].

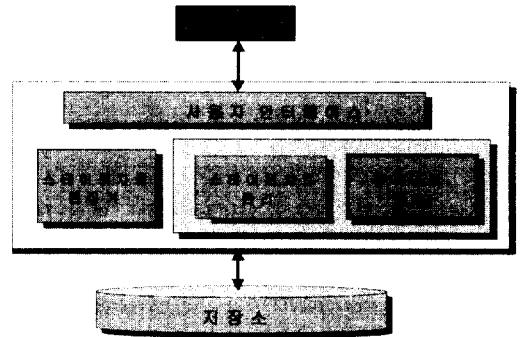
그러나, 기존 도구에서 복사를 통한 Statechart 재사용은 시스템 설계자가 복잡한 기존 시스템 설계도에서 재사용 할 수 있는 State를 찾기 위해 세밀한 분석이 필요하다. 이러한 기존 시스템 설계분석은 시스템 설계가 복잡하면, 복잡할수록 어렵게 된다. 또한 찾아낸 State를 새로운 시스템 설계에 추가하는 것도 쉽지 않을 수 있다.

본 논문에서 기술한 컴포넌트를 통한 State 재사용은 재사용 가능한 State를 컴포넌트로 만들어 저장소에 저장하여 새로운 시스템 설계시 이미 저장된 컴포넌트를 재사용 할 수 있으므로 설계 시간 및 노력이 단축될 수 있다.

3. Statechart 도구 구성 및 구현

3.1 도구 구성

도구는 컴포넌트를 지원하는 Statechart 개발 도구로써 크게 다섯 부분으로 구성되며 [그림 1]과 같다. 구현한 도구는 사용자 인터페이스와 Statechart를 작성, 수정 및 삭제할 수 있는 Statechart 편집기, Statechart 편집기에서 기술한 Statechart의 명세를 저장하고 관리하는 Statechart 관리기, 재사용을 위한 컴포넌트를 생성하고 유지 및 관리하는 컴포넌트 관리기, 마지막으로 생성된 Statechart 및 컴포넌트를 저장하는 저장소로 구성된다. 주요 클래스들은 사용자 인터페이스를 제공하는 StatechartEditorFrame 클래스, Statechart를 기술하는 EditorCanvas 클래스, 기술된 Statechart의 명세를 저장하는 ObjectData 클래스, 그리고 컴포넌트를 생성, 관리하는 ComponentManagement 클래스, 본 논문에서 제안한 컴포넌트의 명세를 가지는 Component 클래스이다.



[그림 1] 구현된 Statechart 도구의 구성

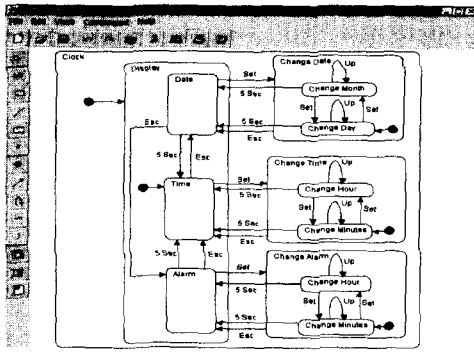
3.2 도구 구현

제안한 도구는 JDK 1.2.2와 Win98을 기반으로 구현하였으며, 도구 실행환경은 Java2를 실행할 수 있는 환경이면 된다.

구현 결과는 시계를 예제를 통해 기술한다. 시계는 각각의 기능이 명확한 부품들로 이뤄져서 작동하므로 각 부품들을 컴포넌트 시키기 적합하며 본 논문은 시계의 날짜, 시간, 알람 등을 설정(setting)하는 부분을 컴포넌트 시켰다. 시계에서 설정부분은 들어오는 입력 값을 사용자가 원하는 대로 맞추는 부분과 그것을 돌려주는 부분으로 구성되어있는 공통의 성질을 가지고 있어서 재사용성에서 컴포넌트 장점을 살릴 수 있기 때문이다.

구현된 도구의 전체 화면은 [그림 2]와 같다. 도구 상단에는 메인 화면과 메뉴 및 툴바 그리고 일반적으로 쓰이는 편집기능이 있다. 도구 왼편에는 Statechart를 작

성할 수 있는 버튼과 컴포넌트를 생성, 유지, 관리하는데 사용하는 MakeComponent, ListComponent, ViewComponent 버튼이 있다. 도구에서 왼편 Statechart 객체에 해당하는 버튼을 누르고, EditorCanvas에 클릭을 하거나 드래그를 하면 Statechart 객체를 화면에 표시할 수 있다. 화면에 표시된 Statechart 객체는 Statechart 객체의 명세를 저장하는 객체를 생성한 후 ObjectData 클래스에 추가된다.

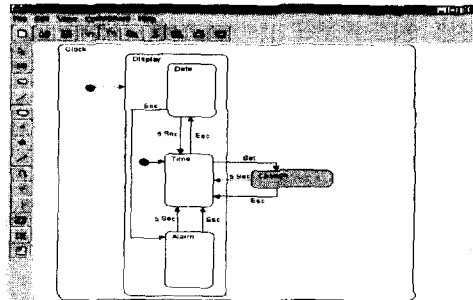


[그림 2] 시계에 대한 Statechart

처음 Statechart 객체중의 하나인 상태를 생성하기 위해 상태 버튼을 누른 후 EditorCanvas에 클릭하면 화면에 상태 객체가 하나 표시된다. 그리고 상태 객체에 해당하는 State Class의 인스턴스가 생성한후 EditorCanvas 클래스가 가진 ObjectData 클래스의 인스턴스에 추가된다. 그 다음 상태 객체를 선택하고 더블클릭하게 되면 상태에 대한 상세한 내용을 기술할 수 있는 대화상자가 생성되고 사용자는 여기에 기술하게된다. 이렇게 기술한 내용은 ObjectData 인스턴스에서 해당 객체를 가져온 후 그 객체에 기술된 내용을 저장하고 ObjectData에 넣어 그 명세가 도구를 사용하고 있는 동안 유지되도록 한다.

컴포넌트를 만들기 위해서 MakeComponent 버튼을 누른 후 화면에 컴포넌트로 만들 부분인 Change Time 상태를 드래그하여 사각형을 만들면 MakeComponent 대화상자가 활성화되어 컴포넌트를 생성시킬 수 있다. 또한 이렇게 생성된 컴포넌트는 ListComponent를 통해서 볼 수 있다. ListComponent에 저장된 컴포넌트를 선택하여 EditorCanvas 클래스에 클릭하면 컴포넌트가 화면에 표시된다. Change 컴포넌트와 기존 상태인 Time 상태를 변이로써 연결하면 Change 컴포넌트에 정의되어있는 인터페이스중에서 하나를 선택한다[그림 3]. 이와 같이 기존 Statechart와 새롭게 추가한 Change 컴포넌트를 연결하

여 새로운 시스템 설계를 작성할 수 있다.



[그림 3] 컴포넌트와 상태간에 인터페이스로 연결

4. 결론

내장형 시스템이 복잡해질수록 가상 프로토타입핑의 중요도가 증가될 것이다. 가상 프로토타입핑은 제품의 기능 및 외관을 컴퓨터를 통해 미리 평가함으로써 설계 상 오류를 제품완성 전에 검증할 뿐만 아니라, 신뢰도가 높은 제품을 생산할 수 있다.

가상 프로토타입핑의 요소 기술 중 정형 명세언어 기술인 Statechart는 제품의 기능적 특성을 나타내는 기술이다. 기존 Rapid PLUS나, Magic Draw등에서 Statechart 재사용은 Statechart를 복사해서 사용하고 있다.

본 논문에서는 제시하고 있는 컴포넌트를 이용한 Statechart 재사용은 새로운 시스템의 설계시 이미 컴포넌트로 구현된 정형 명세를 최대한 재사용 함으로 시스템 설계 시간을 보다 효율적으로 단축시킬 수 있는 장점을 지닌다.

5. 참고문헌

- [1] Hennessy and Patterson, Computer Architecture 2nd edition, 1997
- [2] G. Booch, J. Rumbaugh, I. Jacobson, The Unified Modeling Language User Guide, Addison-Wesley, 1999
- [3] D. Harel, "STATECHARTS : A VISUAL FORMALISM FOR COMPLEX SYSTEMS", Science of Computer Programming 8, 1987
- [4] D. Harel and A. Naamad, "The STATEMATE Semantics of Statecharts", Published in ACM Trans. Soft. Eng. Method, October 1996
- [5] <http://www.e-sim.com/embed/index.htm>
- [6] <http://www.nomagic.com/magicdrawuml/>
- [7] <http://xjtek.com/products/xjcharts/index.html>