

## Virtual Prototyping 이란?

이 희 용, 최 진 영

고려대학교 컴퓨터학과

### I. 서 론

컴퓨터의 발전으로 대형 과학계산으로부터 소형 가전 제품에 이르기 까지 그 이용도가 다양해지고 있다. 특히 컴퓨터 내장형 시스템(embedded system)의 이용도는 범용 마이크로 프로세서 및 주문형 반도체의 발전으로 산업 전반으로 확장 되고 있다. 한 예로 많은 가전 회사들이 가전 제품을 컴퓨터 내장형 시스템으로 개발하여 소비자의 신뢰도와 편리성을 증대하여 부가 가치를 확대하고 경쟁력을 확대하고 있다. 그런데 이러한 산업 전반에서 나타나고 있는 컴퓨터의 이용도가 확대됨에 따라 제품의 수명 주기(lifecycle)는 점차 짧아지고 있다. 한 회사의 새로운 제품은 경쟁사의 새로운 기능을 첨가한 제품 때문에 시장성을 잃고 더욱 편리한 기능을 첨가한 제품을 만들도록 요구하게 된다. 기획, 설계(외관 및 내장기능), 프로토타입 구현, 소비자의 만족도 조사, 생산으로 나누어지는 현재 사용중인 제품의 개발 방식은 점차 짧아지는 제품의 수명 주기로는 그리 적합하지가 않다. 프로토타입이란 제품으로 생산하기 전에 만들어진 실제 크기 아니면 축소형의 실물의 모델을 말한다. 이러한 프로토타입은 제품으로 출고하기 전에 그 제품의 외형 및 기능을 시험하여 시장 경쟁력을 갖추기 위하여 중요한 생산 과정 중의 하나로 사용 되고 있다. 그러나 고전적인 프로토타입의 모델은 실제로 외형을 만들며 기능에 필요한 소프트웨어 및 하드웨어를 만들어야 하므로, 제품의 개발 시간이 길어지며, 비용이 크다는 단점이 있다. 컴퓨터 과학 기술의 발전으로 컴퓨터에만 데이터의

형태로 존재하는 모델이 가능하게 되었으며 이러한 컴퓨터에 의해 만들어진 프로토타입은 실물의 프로토타입을 대치할 수 있게 되었다. 이렇게 컴퓨터에 의해 가상으로 만들어진 프로토타입을 가상 프로토타입(Virtual Prototype)이라 하며, 가상 프로토타입을 만드는 과정을 Virtual Prototyping 이라고 한다.

가상 프로토타입이 실물의 프로토타입을 대치하기 위해서는 현실과 비슷한 외형을 관찰할 수 있어야 하며, 외형에 붙어있는 각종 장치 및 기능이 실제와 비슷하게 시뮬레이션(simulation)되어야 한다. 컴퓨터 모델링을 거쳐서 어떤 형상으로 표현되고, 그런 형상에 렌더링(rendering)이라는 과정을 통하여 색깔을 주면 비록 컴퓨터 속에만 존재하지만 실물과 같은 착각을 느낄 정도의 표현이 가능하다. 시각적으로는 이 정도면 실물같이 보일지 모르지만, 그 제품이 지녀야 할 기능을 시험하자면 한 단계 더 나아가야 한다. 정형 명세기법으로 표현된 제품의 기능을 시뮬레이션이라는 과정을 거쳐서 역시 컴퓨터 스크린에서 작동하게 보여주면 꽤 실물을 보는 듯한 착각을 느끼게 할 것이다. 이러한 요구사항을 위해 Virtual Prototyping은 컴퓨터 그래픽스(또는 가상현실) 기술과 정형기법 기술이 필수적으로 필요하게 되며, 부가의 컴퓨터 기술을 혼합하여 실제 프로토타입을 대치하는 가상 프로토타입의 이용도를 증가시킬 수 있게 되었다. 2절에서는 Virtual Prototyping의 광범위한 이용도에 대해 설명을 하며, 3절에서는 Virtual Prototyping을 위한 기반 기술에 대해 간략하게 소개하기로 한다. 4절에서는 Virtual Prototyping의 응용 사례 및 예를 살펴보고, 5절에서 발전 방

향과 결론으로 끝을 맺기로 한다.

## II. Virtual Prototyping의 이용도

위 절에서 설명된 가상 프로토타입은 기존의 프로토타입 보다 더 넓은 이용도를 가지고 있다. 기존의 프로토타입의 개발은 외형의 디자인이 바뀌면 새로운 프로토타입을 만들어야 하는 단점이 있다. 그러나 가상 프로토타입은 컴퓨터 그래픽스의 모델링 방법을 사용하기 때문에 항상 손쉽게 실시간으로 바뀐 디자인을 확인해 볼 수 있다. 이러한 가능성은 여러 형태의 가상 프로토타입을 만들어 소비자에게 가장 친근감이 가는 디자인의 제품을 생산하는데 편리하게 해 줄 수 있다. 비슷하게 기능이 정형 명세로 표현되어 외형과 연결이 되어 있으므로 새로운 기능의 첨가 -- 즉 새로운 스위치를 첨가하던가 아니면 삭제하는 등의 -- 가 손쉽게 이루어 질 수 있다. 이러한 정형명세는 시뮬레이션 해 볼 수 있으므로 정의된 기능과 외형 디자인이 얼마나 잘 어울리는 지 테스트해 볼 수 있게 된다. 또한 기능의 변환은 실제의 하드웨어나 소프트웨어의 변환이 아니고 정형 명세상에서 일어나는 가상의 것이므로 손쉽게 여러 기능을 첨가, 삭제하며 소비자가 원하면 적정한 기능이 있는 제품을 디자인 하는 데 도움이 된다. 또한 정형 기법을 사용, 제품 기능의 요구가 충족을 하는 지를 검증할 수 있게 된다. 즉 설계상의 오류를 가상 프로토타입의 완성전에 검증함으로써 실제제품에 포함될 수 있는 기능상의 오류를 방지하여 신뢰도가 높은 상품을 생산할 수 있도록 해준다. 실제로 프로토타입을 만든 후에 기능의 오류가 확인이 되어 다시 처음부터 공정을 다시 시작하면 많은 시간을 소모하게 되며, 이러한 시간의 낭비는 제품의 기획으로부터 생산까지의 시간을 상당히 길게 만든다. 이렇듯 가상 프로토타입은 실제제품의 외형 및 기능의 변환을 손쉽게 테스트 해 볼 수 있게 하므로 제품의 생명주기(lifecycle)를 상당히 줄일 수 있다. 실제제품이 생산될 때 필요한 것은 역시 사용자 설명서이다.

이러한 설명서는 제품의 각 부품과 각 부품이 하는 기능을 설명하여야 하는데, 가상 프로토타입이 정형 명세로 이루어져 있으므로 설명서 작성자는 설계자가 원하는 기능을 애매모호함 없이 정확하게 인식하여 설명서를 만들 수 있도록 해 준다. 사용 설명서가 읽고 이해하기 어려웠던 이유중에 하나가 설명서를 준비한 사람이 설계자의 뜻을 정확하게 파악하지 못하고 작성했기 때문이다. 그러므로 외형 디자인과 연결이 되어 있는 정형 명세는 설명서 작성에 큰 도움을 준다.

또한 가상 프로토타입의 이용도는 실물의 프로토타입의 경우보다 광범위하다. 실물의 프로토타입이 만들어진 경우, 장소와 거리의 제약으로 프로토타입을 사용하여 얻게 될 테스트 그룹의 크기가 한정되어 있지만, 가상 프로토타입을 만들 경우 인터넷이나 디스켓 또는 CD의 도움으로 장소와 거리의 제한 없이 많은 수의 테스트 그룹을 형성하여, 제품에 대한 소비자의 반응을 정확히 feedback 시킬 수 있게 되어 소비자의 요구에 맞는 경쟁력 있는 제품을 생산할 수 있게 된다.

제품의 생산시 외관 디자인이 컴퓨터로 되어 있으므로 CAM(Computer Aided Manufacturing) 기술을 이용하여 자동적으로 생산 가능하며 정형 명세로 구현이 되어 있는 제품의 기능은 코드로 자동 생성되던가 아니면 소프트웨어-하드웨어 codesign 기법을 이용하여 원하는 하드웨어의 설계 코드 및 이에 맞는 소프트웨어 코드를 자동생성하게 된다. 이러한 과정은 기존의 프로토타입에서 실제제품을 만드는데 걸리는 시간을 단축시킬 수 있게 된다. 또한 모든 것이 생산자의 표준에 맞게 컴퓨터 모델로 되어 있으므로 새로운 모형과 기능을 첨가한 신제품을 만드는 시간이 줄어들게 된다. 즉 제품의 수명 주기(lifecycle)를 절대적으로 감축할 수 있다.

실물의 프로토타입은 일단 제품의 생산후에는 그 필요성이 감소하는 데 비해, 가상의 프로토타입은 생산후 소비자와의 관계에서도 많은 장점이 있게 된다. 일단 만들어진 제품의 가상 프로토타입은 사용자의 교육용 자료로 사용가능하다. 즉 기존의 사용 설명서는 정적인 단순 그림과 설명으로 이루

어져 있는 데 비해서, 가상 프로토타입을 설명과 함께 디스켓, CD, 또는 인터넷등을 통하여 제공함으로써 3차원 그래픽스와 시뮬레이션을 통한 동적인 설명을 통한 사용자 교육이 가능하며, 소비자의 만족등을 증가시킬 수 있을 뿐만 아니라, 제품의 부가가치 및 경쟁성을 증대시킬 수 있다. 이러한 제품과 소비자 간의 교류는 생산전에 소수의 모니터를 통하여 제품의 의견을 모으는 데에 비하여 다수의 소비자의 직접적인 의견을 수집할 수 있는 기회를 제공하여 주며 이로 인하여 소비자의 요구 사항 및 개선사항을 수집, 새로운 제품의 설계 및 상품화에 많은 도움을 줄 수 있다. 즉 소비자의 욕구를 만족하는 신제품의 신속한 설계와 생산이 가능하다. 그 이용도는 가상 상점에서도 나타난다. 지금까지의 가상 상점에서는 제품의 외관만을 보여 주었는데 비해, 가상 프로토타입을 이용하여 전시된 가상 상점의 제품은 소비자가 직접 제품의 각종 기능을 시험해 봄으로써 선택의 폭을 넓힐 수 있게 된다. 이러한 Virtual Prototyping 기능은 생산업체가 표준기술로 사용함으로써 주문형 상품의 제작이 가능하게 된다. 즉 제품 생산 회사에서만 상품의 설계를 할 필요가 없게 된다. 즉 어떤 소규모 회사에서 자신의 제품을 Virtual Prototyping으로 기능 및 외관을 설계하고 생산 회사에서는 설계된 외관 디자인 및 기능 명세에 따라 자동화된 생산이 가능하게 된다.

### III. Virtual Prototyping을 위한 요소 기술

본 절에서는 Virtual Prototyping을 위한 요소 기술을 설명한다. 위에서 설명하였듯이 Virtual Prototyping은 기본 개념이 컴퓨터 그래픽스와 정형 기법이지만, 정도의 차이는 있지만 컴퓨터학의 여러 분야 기술을 요구한다.

#### 1. 가상 프로토타입의 가시화

Virtual Prototyping에서는 설계된 상품의 작동을 시뮬레이션을 이용하여 검사할 수 있도록 각각

의 기능을 담당하는 가상객체들의 작동을 시각화함으로써 동적인 면을 쉽게 볼 수 있도록 한다. 시각화를 위해서는 우선 상품의 전체적인 겉모양을 위한 모델링, 색상과 표면 재질 등을 표현하기 위한 렌더링, 동적인 면을 표현하기 위한 애니메이션, 그리고 실제감을 표현하기 위한 가상현실등의 그래픽스 기본 기술이 필요하다. Virtual Prototyping의 시뮬레이션을 시각화 하기 위하여 가장 먼저 해야 하는 일은 프로토타입의 형상을 만드는 모델링이다. 모델링에서는 흔히 디스플레이 하는데 시간이 적게 걸리는 wireframe의 형태로 객체의 겉모양을 나타낸다. 물체를 표현하기 위한 방법들은 요구되는 분야에 따라서 여러 가지가 존재한다. 예를 들어 솔리드 모델링의 경우, CSG (Constructive Solid Geometry)라고 하는 객체 표현방법이 사용된다. 객체의 표현방법은 렌더링 알고리즘에 따라서도 바뀌어 질 수 있는데 렌더링 방법중의 하나인 ray tracing에서 구의 표현은 중심점과 반지름의 정보만 있으면 표현할 수 있다. 기하학적 모델링을 위하여는 polygonal representation, bicubic parametric patch, CSG, 그리고 space subdivision technique 등의 방법을 이용할 수 있다.

이러한 모델링 방법을 이용하여 상품의 전체적인 모양이 만들어지면 렌더링을 통하여 실제와 가까운 정도의 이미지를 얻을 수 있다. 렌더링에서는 빛의 특성, 물체의 표면이 가진 특성 등을 이용하여 모델링을 통해 만들어진 겉 모양에 색을 입힌다. 상품의 복잡한 표면을 표현하기 위해서 texture mapping, bump mapping, 그리고 displacement mapping 등을 이용하기도 한다. 최근에는 빛과 색에 관한 물리적인 특성을 고려한 ray tracing, radiosity 등의 렌더링 방법이 각광받고 있다. 렌더링이 끝나면 이제 실제 상품과 거의 같은 이미지를 얻을 수 있다.

Virtual prototyping을 이용하여 상품의 작동을 실험하기 위해서는 실제와 같은 겉모양도 중요하지만 동적인 면, 즉 상품의 실제 동작을 보여줄 수 있어야 하는데 이것을 위하여 애니메이션이 필요하다. 애니메이션은 kinematics, dynamics등을 이

용하여 물체의 움직임을 표현하거나 모델의 위치와 물체표면의 속성, 물체를 바라보는 시점등을 바꾸어준 후 렌더링을 하는 과정을 반복함으로써 가상 프로토타입의 동적인 면을 보여줄 수 있다. 최근에는 렌더링 과정에 걸리는 시간을 줄이기 위하여 image morphing을 이용한 image-based rendering을 이용하여 빠른 시간에 애니메이션을 수행하는 기법이 사용되기도 한다.

지금까지 설명한 모델링과 렌더링, 그리고 애니메이션을 이용하여 상품의 실제 동작을 시각화하여 보여줄 수 있다. 이제 이러한 virtual prototyping을 실제로 사용해 보고 작동시키며 사용자에게 상호작용을 통한 실제감을 주기위하여 가상 현실의 기술을 이용할 수 있다. 가상현실이란 컴퓨터가 만들어낸 가상의 세계를 사용자에게 다양한 감각채널을 통해 제공함으로써 사용자로 하여금 이 가상세계에 몰입하도록 하는 동시에, 가상세계 내에서 현실 세계에서와 같은 자연스러운 상호작용을 가능하도록 하는 제반기술과 이러한 기술에 필요한 이론적 바탕을 지칭한다. 가상현실 시스템이 사용자에게 몰입감을 주기 위해서는 여러 가지 고려해야할 문제들이 존재하는데 특히 실시간 렌더링 기법이 요구된다. 초당 20 frame 이상의 영상을 생성하면서도 가능한 사실적인 표현을 해야 하는 실시간 렌더링을 위해서는 가상세계를 구성하는 모델링의 복잡도를 줄여야 하는데, 이를 위해서 efficient surface tessellation, level of detail, visibility culling, 그리고 image의 coherence를 이용한 image-based rendering 등의 방법을 이용할 수 있다.

Virtual Prototyping이 제공하는 시각화를 통하여 고객은 실제로 제품을 사용하는 것같은 경험을 할 수 있다. Virtual Prototyping은 컴퓨터 스크린을 통해서 사용자들에게 보여지게 되는데, 사용자들은 이 프로토타입을 시뮬레이션을 통해서 사용해 보게 되며 그 결과에 대한 의견을 제공한다. 이 결과로 얻어진 여러 가지 의견들을 통해서 가상 프로토타입은 더욱 발전된 모양으로 바뀌게 된다.

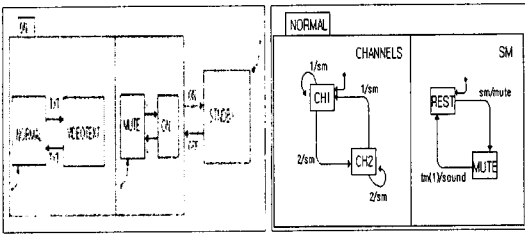
## 2. 가상 프로토타입의 기능

Virtual Prototyping에서의 정형 명세는 만들고자 하는 시스템의 기능 명세뿐만이 아니라, 사용자 설명서에 이르기까지의 전 생산과정에서 핵심적인 역할을 하게 되는 명확하고 정확한 기능 명세이어야 한다. Virtual Prototyping에서 사용되는 명세는 한가지 방법만이 존재하는 것이 아니고 Virtual Prototyping에 사용 가능한 여러 가지 기법이 존재할 수 있다. 시각적인 정형 명세 기법인 Statechart[HP85]와 Statechart의 변형인 Statemate[i-Lo87] 및 Modechart[JM89], GCSR[ALC95] 등이 사용될 수 있다. 또한 텍스트 형태의 정형명세인 CCS, CSP, ACSR 등도 이용 가능하다. 본 절에서는 Harel의 Statechart에 대해 간략히 알아보기로 한다.

Statechart는 그 반응시스템의 행동을 명세하는데에 적당한 formalism의 하나로서 제공된다. 그 근본 구조는 State Transition Diagram(STD)으로 부터 세가지 중요한 요소를 포함한 extension으로 생각할 수 있다. 여기서 세가지 요소는 hierarchy, concurrency, communication인데, 이를 통해 STD보다 훨씬 효율적이고 구조화된 새로운 명세도구로서 사용할 수 있게 된다. Statechart의 등장 배경을 살펴보면, 전통적인 transition diagram에서는 커버되지 않는 개념이 하나 있는데, 그것은 바로 structure이다. 기술되는 시스템이 복잡해지면, 정보를 구조화하고 요약정리하는 방법이 없기 때문에 diagram은 박스와 화살표등으로 매우 혼란스럽고 복잡한 형태가 된다. Harel은 hierarchical state를 이용하여 structure를 표현하는 명백하고 효과적인 방법을 개발하였다. Hierarchy(Depth)는 tree에서 subtree가 파생되듯이, 하나의 state가 substate를 가질 수 있게 하는 것으로 계층적 구조를 표현하는데 필수적이다. Concurrency(Orthogonality)는 state들 간에 병렬성을 제공한 것인데, 어떤 시스템이 두 개 또는 여러 개의 subsystem으로 나뉘어져 서로 cummunication하면서 동시에 수행되어 질 때, 이를 표현해주기 위한 것으로, 여러 개의 parallel component가 모여서 하나의 state를 구성하게 되며, 각 component 역시 하나의 state이다. 마지막

으로 Communication이라는 것은 reactive system에서의 각 component들 사이에 어떤 식으로든 communication을 하게 되는데 이를 위해 transition을 trigger, label, 그리고 action의 세가지로 구성하여 trigger는 transition을 위한 조건, label은 transition에 붙이는 이름, action은 trigger의 조건이 만족되었을 때 발생하는 사건으로 각각 분리하여, transition이 취해졌을 때 발생하는 이벤트가 broadcast되어 이를 다른 transition에서 감지하여 새로운 trigger로서 작용할수 있도록 하는 방식으로 communication을 하게 된다.즉, statechart는 다음과 같은 요소들로 구성되어 있다고 표현할 수 있다.

Statechart = STD + hierarchy + concurrency + communication



다음은 Statechart의 한 예이다.

Reactive system의 예로서 리모콘에 의해 작동되는 TV수상기 시스템을 살펴보기로 한다.우선 시스템은 리모콘의 버튼을 누름으로써 Input event가 생성되며, 그로인해 시스템은 이벤트를 보내고 state를 바꿈으로써 반응하는 형식이다.시스템을 활성화시키는 on 버튼이 있으며, standby 모드로 돌아가게 하는 off 버튼이 있다.On 상태에 있을 경우, normal broadcast mode나 videotext mode (teletext)의 둘중에 하나의 상태에 있게 되며, videotext mode에 있을 경우, 특별한 text page들을 보여주며, 이는 TV프로그램과 같이 보내어 진다.두 mode를 switching(toggling)하는 것은 txt라는 버튼에 의해서 수행된다.그 행동을 기술하는 submachine은 on state의 안에 그려지게 되며, 이러한 구성은 어느 상태에서도 적용가능하다.On이라는 것은 normal이거나 videotext라는 것을 의미

하므로 on state와 같은 state를 OR state로 부른다.점으로부터의 화살표는 더 윗 단계의 state가 왔을 때 그 state의 substate들 중에 설정되는 state를 specify해주는 기능을 한다.위의 경우, sound와 image operation은 각각 서로 독립적이다.즉, normal mode에서 videotext mode로 바꾼다고 해도 sound에는 영향을 미치지 않는 것이다.따라서 state on을 sound와 image의 두 개의 orthogonal substate를 이용해 구성한다.간단명료하게 하기 위해 일단은 sound level을 mute와 on의 두가지로 구분하며, 각 state로의 스위칭은 mute button과 sound button으로 이루어 지며, 물론 설계시 sound level을 좀더 여러 단계로 구분할 수 있다.

일반적으로, orthogonal component들이 서로 완벽히 독립적인 경우는 거의 없다.따라서 그 상호 영향과 관계, 통신을 명세해 줄 필요가 있게 되며, 트랜지션에 output을 줌으로써 이를 가능하게 한다.이 output은 시스템에서 언제든 sense되어 다른 트랜지션에 trigger할 수 있는 broadcast이다.위의 예에서, 채널을 다른 채널로 바꿀 때, 보통 1초정도 sound가 꺼지게 된다.이것을 모델하기 위해 channels와 sm(switching mute)두 개의 orthogonal component들을 추가한다. 리모콘의 channel button을 누르면, 그 channel로 switching이 일어나며, sm이라는 이벤트가 생성된다.그 이벤트는 state sm에서의 transition에 의해 mute라는 이벤트가 생성되도록 하며 sound가 꺼지게 한다.그리고 1초후, 다시 음을 켜기위해 sound라는 event가 생성되는데, 이는 time-out이라는 이벤트(tm)에 의해 가능하게 된다.즉, mute state에 들어왔을 때 counting을 시작하여 clock event를 계속 세다가 값이 1에 이르면(1초가 흐르면) generate되는 카운터의 역할을 하는 것임을 알 수 있다.Transition의 레이블은 두 부분으로 구성이 되는데, trigger와 action이 바로 그것이다: trigger는 transition이 언제, 어떠한 상황에서 일어날 것인지에 대한 정보이며, action은 transition이 일어 남으로써 발생하는 event들이다. Transition에서의 trigger는 event들의 logical

proposition의 형태가 될 수도 있다. 즉,  $(\neg a \wedge b / e)$ 와 같은 transition은, 동시에 a는 absent하고 b는 present해야만 발생한다는 것을 말한다.

### 3. 가상 프로토타입의 제품화

Virtual Prototyping은 최종 상품을 위한 하드웨어와 소프트웨어의 코드를 자동적으로 생성해 주기 위해 사용된다. 이러한 작업은 사용된 정형기법을 입력으로 한 하드웨어-소프트웨어 codesign 기법을 사용하게 된다. 즉 정형기법으로 서술이 된 명세를 이용하여 시간 및 하드웨어 가격등을 고려하여 적절하게 나누어진 하드웨어 설계 언어인 VHDL [이대영95] (꼭 VHDL 일 필요는 없고 환경에 맞는 언어이면 된다) 코드와 구현된 하드웨어에서 돌아갈 소프트웨어 코드(어셈블리 언어 또는 C 언어등)로 자동 생성하게 된다. 물론 이러한 codesign 기법은 아직 연구 단계이고 정확한 결과를 내기에는 지금은 힘들지만 범용 마이크로 프로세서를 가정할 때에는 codesign을 생략하고 명세에서 코드를 직접 생성하는 연구는 상당히 진척이 되어 있다.

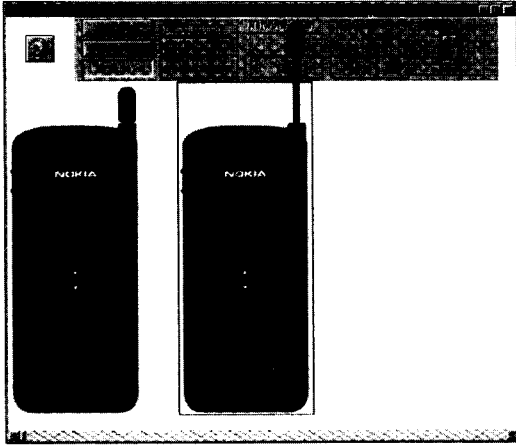
### 4. Virtual Prototyping을 위한 객체지향 소프트웨어 디자인

제품개발의 첫 번째 단계는 개념의 정립과 기능설계를 하는 것이다. 제품의 명세에 대한 개념과 생각은 마케팅, 엔지니어링, 관리 등으로부터 얻어지게 된다. 이렇게 해서 만들어진 개념들은 가상 객체들로 구성된 virtual prototype으로 변환된다. Virtual prototype을 개발하기 위한 핵심기술인 객체중심 부품기술(object-oriented component technology)은 객체의 재사용과 완벽한 커스터마이징(customization)을 지원하며 Virtual Prototyping을 빠르고 간단하게 구현할 수 있도록 한다. 일반적으로 객체지향 기법은 모듈화, 캡슐화, message passing, 상속성, 그리고 다형성등의 특성을 제공하기 때문에 객체지향 기법으로 설계된 시스템은 사용자에게 확장의 용이성을 지원해 주며 사용자에게 유용성을 제공하고자 하는 시스템에 많이 적용된다. Virtual Prototyping을 구성하는 가상객체

들은 시뮬레이션을 위한 기능(functionality)과 함께 시각화 요소(visual component)들을 가지고 있으며 target system의 물리적인 객체로 mapping될 필요성 때문에 각 객체를 표현하는 코드 라이브러리도 포함할 수 있다. 시스템의 요소들 모두를 객체를 사용하여 표현하고 시스템의 모든 상태에서의 activity를 객체들의 함수(function)로서 표시함으로써 시스템(embedded system)의 제어, 데이터의 흐름(data flow), 그리고 입출력 인터랙션(I/O interaction)을 모두 포함한 완전한 프로그램 코드를 자동으로 생성할 수 있다. 개발자는 이러한 가상 객체들을 그래픽스를 통하여 조작(graphics manipulation)하여 GUI 환경에서 가상 프로토타입을 쉽게 개발할 수 있다. 객체지향 소프트웨어 공학을 통하여 modular한 component(object 또는 metaobject)들이 이루어지면, 새로운 시스템을 구성할 때 여러 가지 component들을 mix하고 match하여 우리가 고안해 내고자 하는 새로운 제품을 이미 존재하는 component들을 최대한으로 이용함으로써 디자인의 재사용을 피할 수 있을 것이다. 기능이 흡사한 여러 model이 한 종류의 제품에서 생산되는 것이 보통인 일인데 그런 점들을 Virtual Prototyping을 이용할 때에도 역시 그대로 고려해야 할 것이다. 객체지향 소프트웨어 디자인 기법중 이런 점을 특히 고려한 것으로 Open Implementation[MLMK97]을 들 수 있으며, Virtual Prototyping에도 적용하면 좋은 결과를 얻을 것으로 예상된다.

## IV. Virtual Prototyping의 응용분야 및 응용사례

최근의 산업에서는 새로운 제품의 개발 주기가 예전에 비해서 훨씬 빨라졌다. 이를 기업의 입장에서 보면 새로운 제품을 얼마나 빨리 시장에 내놓을 수 있는지가 성공의 관건이라고 할 수 있다. 그러나 이러한 기업의 입장과는 달리 제품의 복잡도는 하루가 다르게 높아지고 있다. 따라서 이와 같은



〈그림 1〉 Rapid PLUS를 이용한 휴대폰 개발 화면

문제를 해결하기 위한 방법이 절실히 요구되고 있다.

Virtual Prototyping은 위에서 언급한 문제를 쉽게 해결해 줄 수 있다. 기존에 사용되던 제품의 개발 방법은 기획, 설계, 제작, 마케팅, A/S 등의 작업들이 순서적으로 일어났었다. 그러나, virtual prototyping을 이용하는 경우에 이와 같은 대부분의 작업들이 동시 다발적으로 이루어질 수 있다. 따라서 좀 더 빠른 제품 개발을 할 수 있게 된다. 또한 많은 부분이 자동화되고, 실제 제품이 나오기 이전에 미리 테스트와 디버깅을 할 수 있기 때문에 최종 생산물의 품질을 더욱 향상시키는 것이 가능하다.

현재 외국에서는 Virtual Prototyping에 대한 연구가 활발하게 진행되고 있다. 현재 발표되고 있는 연구를 살펴보면 가전 제품 개발을 위한 Virtual Prototyping 시스템[PSS96], 자동차 오디오나 컨트롤 패널의 설계를 위한 시스템[Klei96], 가상 공장 구축을 위한 시스템, 공항의 관제 시스템의 설계와 검증, 마이크로 콘트롤러에 기반을 둔 DSP 시스템 개발[ME95] 등 다양한 분야에 적용되고 있는 것을 볼 수 있다. 특히 가상 공장 구축 등과 같은 시스템은 실제로 공장을 구축하기 이전에 가상으로 컴퓨터 상에 미리 구축된 공장을 분석하여 생산 라인의 최적화나 오류를 최소화할 수 있게 해 줌으로써 비용면에서 상당한 이익이 얻을

수 있다. 또한 이스라엘의 한 회사인 Emultek에서는 statechart에 기반하여 VP 솔루션을 제품화한 Rapid PLUS를 발표하였다[emul97]. 이러한 Virtual Prototyping을 지원하는 툴을 이용하여 개발하는 경우에 의외로 많은 효과를 거둘 수 있다. 제품의 개발을 병렬적으로 진행시킬 수 있기 때문에 제품 개발 시간을 줄일 수 있는 것은 당연하다. 그림 1은 실제로 Rapid PLUS를 이용하여 휴대폰의 virtual prototype을 만든 것을 화면으로 보여준 것이다. 외형이 그대로 화면에 나타날 뿐만 아니라 실제 제품을 사용하듯이 조작해 볼 수도 있다. 그림에서는 두 대의 휴대폰을 이용하여 서로 통화하고 있는 장면이다. 실제 제품을 사용하듯이 번호를 입력한 후에 <snd> 버튼을 눌러 통화하는 모습을 볼 수 있다. 개발자들은 네트워크에 공유된 그림 1과 같은 virtual prototype을 보면서 작업을 하기 때문에 의사 교환이 쉽게 이루어질 수 있다. 또한 이렇게 만들어진 virtual prototype을 실제 고객들에게 선보임으로써 제품이 나오기 이전에 고객들이 원하는 형태로 수정이 가능하다. 또한 기존에는 많은 시간과 노력이 필요하던 제품의 문서화 작업을 자동화하여 준다. 이러한 기능을 이용하여 시스템 개발자는 보다 쉽게 사용자 설명서 등을 제작할 수 있다[emul97]. 또한 Rapid PLUS의 경우에는 이렇게 제작되어진 virtual prototype과 그 문서들이 여러 가지 형식의 출력물로 만들어진다. 특히, 네트워크를 효과적으로 지원하기 위하여 웹 브라우저에서 읽을 수 있는 형식으로 출력된다는 것은 관심을 가질만 하다. 이렇게 작성된 문서는 네트워크에 연결된 곳이라면, 고객이나 고객을 지원하기 위한 고객지원팀 그리고 제품을 판매하는 곳 어디서나 사용이 가능하다.

현재 국내외의 많은 회사에서 Rapid PLUS와 같은 시스템들을 도입하거나 도입을 생각하고 있는 것으로 알려져 있다. 실제로 모토로라와 같은 회사에서는 이러한 Virtual Prototyping 시스템을 이용하여 pilot project를 수행함으로써 제품의 개발 기간을 대폭 단축할 수 있었다는 보고를 하였다.

앞에서 살펴본 Virtual Prototyping 기술들은 현재 급격하게 시장이 팽창하고 있는 통신 장비, 의

료 기기, 자동차 시스템, 가전 제품 등의 개발에 널리 적용될 수 있을 것이다. 이 밖에도 Virtual Prototyping의 또 다른 방향의 추가적인 장점도 존재한다. 작업 과정에서 만들어진 virtual prototype을 인터넷과 같은 가상 공간에 공유시킴으로써 전자 상거래 등에도 이용이 가능하다. 즉, 실제의 제품을 만져보는 듯한 느낌을 받으면서 가상환경에서 물건을 구입하거나 실제로 제품을 구입한 소비자들에게 사용법을 가르쳐주는 방법으로도 virtual prototype을 응용하여 사용할 수 있을 것이다.

## V. 발전방향과 결론

소비자의 구미에 맞는 제품을 생산하여 경쟁성을 높이기 위해서는 보다 신뢰성이 있고, 외장 디자인이 소비자의 구미에 맞는 상품을 다른 경쟁사에 비해 빠른 시간에 만들 수 있어야 한다. 이러한 요구는 현대의 세계적 경쟁시대에 필수로 여겨지고 있으며, 회사의 경쟁력 증가에 큰 영향을 미치고 있다. Virtual Prototyping은 이러한 요구를 충족시켜 줄 수 있을 뿐만 아니라, 인터넷을 통한 소비자 교육 및 매뉴얼 작성 등 여러 방면에서 사용이 될 수 있다. Virtual Prototyping에 필요한 기반 기술은 위에서 설명하였듯이 현재 컴퓨터 과학 기술로 거의 가능하므로 현재 미국에서는 Virtual Prototyping을 위한 도구가 상품화 되어 시장에 나와 있으며 그 이용도는 차츰 그 범위를 넓혀가고 있다. 이 중 Emultek이라는 회사의 제품인 RAPID PLUS는 가전 제품에의 적용이 상당히 용이하며 관심있는 독자는 [www.emultek.com](http://www.emultek.com)을 확인하기 바란다. 이 회사의 사이트에는 간단한 데모 시스템이 제공되어 있어 본 논문에서 설명된 것을 직접 실행해 볼 수 있다. 이러한 Virtual Prototyping은 한 개의 독립된 기반기술로 이루어진 것이 아니라, 여러 가지의 기술이 조화되어 이루어지는 것으로 그 부가가치나 영향이 상당히 크리라 믿어진다. 현재 산업계에서 이용이 될 수 있

는 다양한 Virtual Prototyping 도구가 빨리 구현되어 세계 우수 기업과의 경쟁력에서 뒤떨어지지 않는 기반 도구로 사용 되리라 믿어진다.

## VI. 감사의 글

이 원고를 준비하는 과정에서 몇몇 대학원생들의 도움이 컸기에 감사의 뜻을 전한다. 고려대학교 컴퓨터학과 프로그래밍시스템연구실의 박정규와 신호윤, 그리고 정형기법연구실의 박찬국이 그들이었다. 자료 준비에서 초기 draft를 자세히 읽고 이런 저런 오류를 고쳐주는 등 시간을 아끼지 않았다. 아직도 남은 오류는 두 저자의 책임임을 밝혀둔다.

## 참 고 문 헌

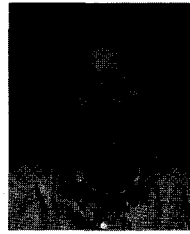
- [1] H. Ben-Abdallah, I. Lee, and J.-Y. Choi, "A Graphical Language with Formal Semantics for the Specification and Analysis of Real-Time Systems," In Proceedings of Real-Time Systems Symposium, 1995.
- [2] <http://www.emultek.com>
- [3] D. Harel and A. Pnueli, "On the Development of Reactive Systems," in Logic and Models of Concurrent Systems, (NATO ASI Series, Vol. 133, K. R. Apt, ed.) Springer-Verlag, Berlin, 1985.
- [4] i-Logic Inc. "The Languages of STATEMATE," in Documentation for the STATEMATE System, 1987.
- [5] F. Jahanian and A. K. Mok, "Modechart: A Specification Language for Real-time Systems," IEEE Transactions on Software Engineering, Nov. 1989.
- [6] Robert H. Klein, "The advantage of



MODEL BEHAVIOUR,” New Electronics, Apr. 1996.

- [8] V. K. Madisetti and T. W. Egolf, “Virtual Prototyping of Embedded MicroControlled-Based DSP Systems,” IEEE Micro, Oct. 1995.
- [9] C. Maeda, H. W. Lee, G. Murphy, and G. Kiczales, “Open Implementation Analysis and Design,” In Proceedings of the 1997 Symposium on Software Reusability (SSR’97), Boston, Mass., 1997.
- [10] P. S. Pulli, M. L. Salmela, and J. K. Simila, “Virtual Prototyping based development and marketing of future consumer electronics products,” IFIP word computer congress, Sep. 1996.
- [11] 이대영, 조원경, 정연모, 오재곤, 하드웨어 설계를 위한 VHDL 기초와 응용, 홍릉 출판사, 1995.

## 저자 소개



### 李 喜 雄

1952年 2月 13日生, 1982年 6月 University of Utah(학사: Computer Science), 1986年 6月 Stanford University(석사: Computer Science), 1992年 6月 University of Utah(박사: Computer Science), 1982年 7月~1983年 9月 Software Engineer, Evans & Sutherland Computer Co., 1983年 10月~1985年 9月 MTS, Sandia National Lab., 1985年 10月~1986年 10月 Research Staff Member, Xerox Palo Alto Research Center, 1986年 10月~1993年 2月 Research Computer Scientist, Evans & Sutherland Computer Co., 1993年 3月~현재 부교수, 고려대학교 컴퓨터학과, 주관심분야: 프로그래밍언어, 분산객체, 가상현실, 지속성객체



### 崔 振 榮

1959年 3月 4日生, 1982年 2月 서울대 컴퓨터 공학과(학사), 1986年 6月 Drexel University(석사), 1993年 8月 University of Pennsylvania(박사), 1987年 9月~1993年 8月 RA, University of Pennsylvania, 1993年 9月~1996年 2月 Postdoc, University of Pennsylvania, 1996年 2月~현재 조교수, 고려대학교 컴퓨터학과, 주관심 분야: 컴퓨터이론, 정형기법, 실시간 시스템