

Animated Simulation 시스템 설계 및 구현

김상필, 배영환
한국전자통신연구원 시스템설계자동화팀
전화 : (042) 860-6547

Design and Implementation of Animated Simulation System

Sang-Pil Kim, Yung-Hwan Bae
System Design Automation Team, ETRI
E-mail : spkim@etri.re.kr

Abstract

In this paper, the animated simulation system (Anisim) is proposed in order to develop an efficient functional system verification tool. It displays the simulation results of the designed system using graphic animation with various models for the target system. With simple interface definitions given by the user, Anisim generates interface codes automatically. Users can describe and model the target system with the generated interface codes. Since the simulation engine is implemented in C-language, modeling is very simple and simulation can be performed in real time.

I. 서론

과학기술이 발달하면서 우리 주변의 전자 시스템도 다양하고 복잡한 형태로 발전하고 있다. 매우 복잡한 시스템을 설계하고 개발하는 사람들이 겪는 어려움 중의 하나는 개발하고자 하는 시스템의 작동 방식이나 성능을 정확히 파악하는 일이다. 현재 일반적으로 사용되고 있는 추상화된 시스템 모델은 전문가를 제외하면 이해하기가 어렵고, 전문가들도 시스템의 성능을 정확히 파악하기가 쉽지 않다. 만일 추상화된 모델이 아닌 실세계에서 사용자가 작동시킬 수 있는 형태로

시스템을 모델링 한다면 시스템 개발에 관련된 모든 사람들이 쉽게 이해할 수 있고 성능을 검증할 수 있을 것이다.

Animated simulation system은 대상 시스템을 그래픽을 이용해서 다양한 형태로 모델링한 후에, 시스템의 시뮬레이션 결과를 그래픽 애니메이션 기능을 통해 시각적으로 나타냄으로써 시스템의 기능을 쉽게 검증할 수 있는 툴이다. Animated Simulation Tool을 사용할 경우 대상 시스템의 작동을 그래픽 애니메이션을 통해 실세계 시스템처럼 시각적으로 나타나므로 시스템을 쉽게 검증할 수 있다.

본 논문에서는 Animated simulation 시스템인 Anisim의 구성과 작동 방식, 그리고 이를 이용한 응용 시스템 구현과 필요한 인터페이스에 대해서 기술한다. 기본적으로 사용자가 주어진 형식에 따라 인터페이스를 간단히 정의하면 Anisim 툴이 인터페이스 코드를 자동 생성하게 된다. 따라서 사용자는 자동 생성된 인터페이스를 이용해서 시스템의 동작을 기술하고 시스템을 모델링하면 된다. 자동 생성되는 인터페이스에 대한 내용은 본문에서 자세히 기술한다.

II. 시스템 구성

2.1 Animated Simulation 방식

이벤트 시뮬레이션 결과에 대한 애니메이션은 시뮬레이션 프로젝트를 수행하는데 있어서 필수적인 요소

로 인식되고 있다. 대부분의 경우 animated simulation 소프트웨어들은 특별한 시뮬레이션 틀이나 시뮬레이션 언어에 종속되어 있다. 하지만 몇 가지 틀은 시뮬레이션 틀이나 애니메이션 도메인과 독립적으로 사용될 수 있도록 만들어져 있다. 그리고 대부분의 틀들은 내부 수행 방식에 따라 그림 1에 표현된 것처럼 세 가지 타입으로 구분될 수 있다.

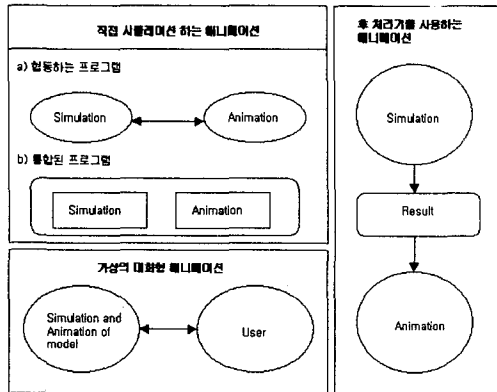


그림 1. Animated simulation 틀의 종류

□ 후처리를 사용하는 애니메이션

가장 단순한 형태로 시뮬레이션 결과를 외부 파일에 저장한 후, 그 파일을 입력으로 받아서 애니메이션을 수행하는 방식이다. 시뮬레이터와 독립적으로 사용할 수 있다는 장점이 있다.

□ 직접 애니메이션하는 시뮬레이션

이 방식은 두 가지 형태가 있는데 하나는 애니메이션 프로그램과 시뮬레이션 프로그램이 서로 통신하면 작업을 수행하는 경우이고, 다른 하나는 하나의 프로그램에 애니메이션과 시뮬레이션 프로그램이 포함되어 있어서 하나의 작업(task)로 처리되는 경우이다.

□ 가상의 대화형 시뮬레이션

앞의 두 가지 경우와는 다르게 수행 도중에 대화형으로 작업을 지시할 수 있기 때문에 실제 시스템과 가장 가까운 형태이다.[1]

2.2 Anisim 시스템

Anisim 틀은 실세계 시스템을 가장 잘 묘사할 수 있는 가상의 대화형 시스템으로 사용자가 주는 이벤트에 따라 해당 작업을 수행하는 방식이다. Anisim은 내부 작업 수행 방식은 server/client 모델로 구성되어 있다. Client 측에는 대상 시스템을 모델링하고 애니메이션하는 Modeler 및 Animator 프로그램이 있고,

server 측에는 사용자가 정의할 시뮬레이션 엔진과 통신할 수 있도록 지원하는 server, 그리고 시뮬레이션 엔진 프로그램의 인터페이스를 자동 생성하는 InterfaceGen 프로그램과 실제 시뮬레이션 작업을 수행할 응용프로그램으로 구성된다. Client 프로그램은 객체지향 언어인 Java로 되어 있고, 나머지는 C-언어로 구현되어 있다. Anisim 시스템 구성을 그림 2에 나타내었다.

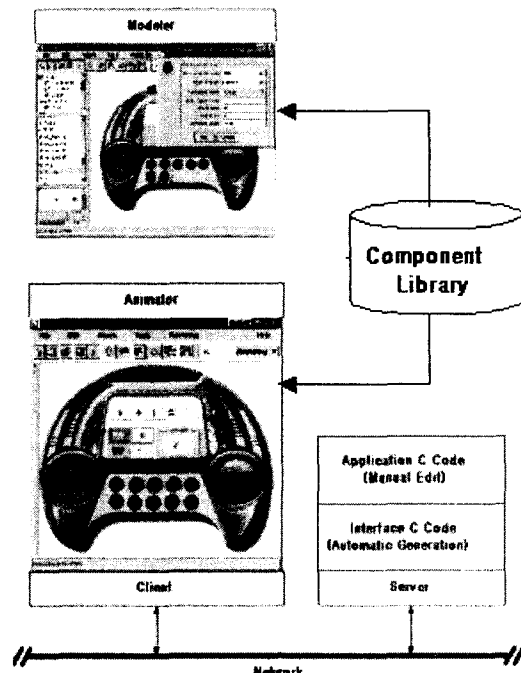


그림 2. Anisim 시스템 구성

III. 동작 원리

Animated Simulation Tool은 server/client 모델로 구성되어 있기 때문에 사용자가 그래픽 이미지 조작을 통해서 client측의 모델링된 시스템을 작동시키면, 그 이벤트에 해당하는 데이터가 네트워크를 통해서 Server측의 어플리케이션(시뮬레이션) 프로그램으로 가서 수행이 된 후, 그 결과를 다시 네트워크를 통해서 client 프로그램으로 보낸다. 그러면 애니메이션 프로그램은 그 결과에 따라 미리 정해진 방식으로 시스템을 작동(애니메이션)시킨다. Client와 서버 사이의 통신을 위한 내부 수행 작업의 순서와 서버와 시뮬레이션 엔진 역할을 하는 응용프로그램간의 관계는 그림 3과 같다. 기본적으로 클라이언트와 서버의 통신은 소켓을 이용한 접속 방식이고 서버와 응용프로그램간의

통신은 두 개의 파일을 이용한 연결 방식이다.

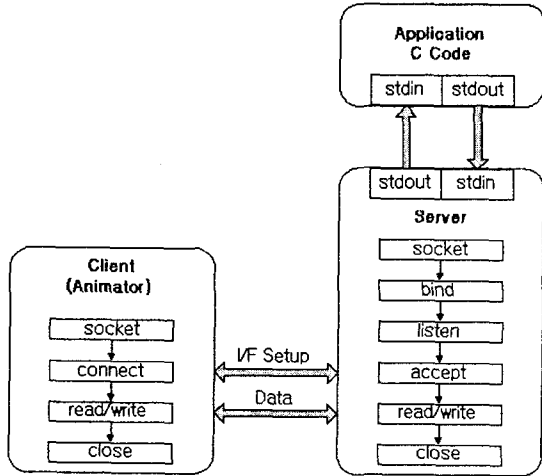


그림 3. Anisim의 작동 원리

IV. Anisim을 이용한 시스템 설계

Anisim 시스템을 이용한 설계는 몇 가지 작업으로 나누어질 수 있다. 우선 사용자는 시스템을 애니메이션 부분과 시뮬레이션 부분으로 분리하고, 둘 사이의 인터페이스를 정의한다. 그런 다음에 각각의 작업은 독립적으로 병렬로 수행할 수 있다. 그림 4에 Anisim을 이용해서 설계하는 흐름도가 정의되어 있다.

유아용 덧셈기 장난감(Flash)을 목표 시스템으로 구축한 사례를 중심으로 Anisim 시스템의 내부 구조를 분석하도록 한다.

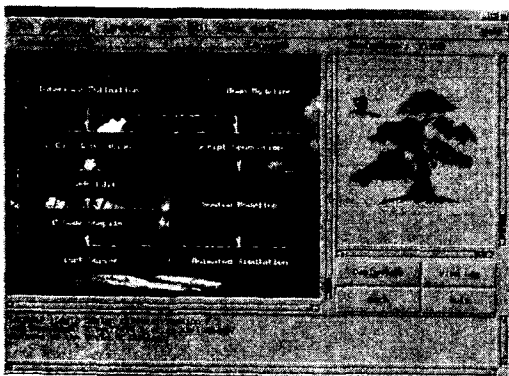


그림 4. Anisim을 이용한 시스템 설계 흐름도

4.1 예제 회로의 사양

'flash' 시스템은 최상위 기능블럭으로 표 1과 같은 입출력 단자를 갖는다. 그리고 'flash' 기능블럭의 회로

는 그림 5와 같이 논리회로도 편집기를 이용하여 나타낼 수 있다.[2]

표 1 'flash' 기능블럭의 입출력 단자

단자이름	비트	입출력	기능
data	5	in	입력되는 5비트 정수 값
dvalid	1	in	정수값이 유효한 시간을 지정
clock	1	in	clock 신호
reset	1	in	reset 신호
count	4	out	덧셈을 맞춘 개수 표시
correct	1	out	덧셈 결과의 옳고 그름표시
answer	5	out	덧셈 결과 출력

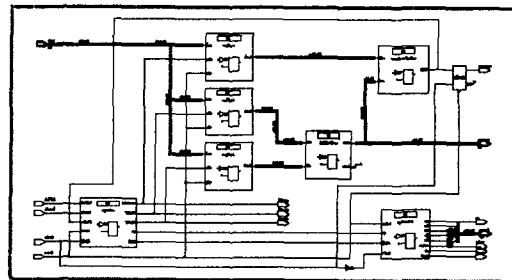


그림 5. 'flash' 기능블럭의 논리회로도

4.2 인터페이스 정의

응용프로그램에 대한 인터페이스를 직접 코딩할 경우 애니메이션 시스템에 대해서 자세히 알아야 하기 때문에 사용자들에게 큰 부담이 된다. 그래서 시스템에 대한 스펙을 간단하게 기술할 수 있는 aif(application interface format) 형식을 도입해서, 사용자가 aif 파일에 시스템의 인터페이스를 간단히 기술하면 InterfaceGen프로그램이 자동으로 인터페이스 프로그램을 생성해주도록 되어 있다. 그림 6은 사용자가 쉽게 인터페이스를 정의할 수 있게 해주는 프로그램의 GUI이다.

	VARIABLE NAME	BEAM_INSTANCE NAME	CORRECTION	PROCEDURE	EDIT ACTION
VARIABLE	DATA	S_DATA	GENERATE	set_data	RESERVE
VARIABLE	DVALID	S_VALID	GENERATE	set_valid	RESERVE
VARIABLE	CLOCK	S_CLOCK	DISPLAY	set_clock	RESERVE
VARIABLE	RESET	S_RESET	DISPLAY	set_reset	RESERVE
VARIABLE	COUNT	S_COUNT	DISPLAY	set_count	RESERVE
VARIABLE	CORRECT	S_CORRECT	GENERATE	set_correct	RESERVE
VARIABLE	OPERAND1	S_OPERAND1	DISPLAY	set_operand1	RESERVE
VARIABLE	OPERAND2	S_OPERAND2	DISPLAY	set_operand2	RESERVE
VARIABLE	RESULT	S_RESULT	DISPLAY	set_result	RESERVE
VARIABLE			GENERATE		RESERVE
VARIABLE			GENERATE		RESERVE
VARIABLE			GENERATE		RESERVE
VARIABLE			GENERATE		RESERVE

그림 6. 인터페이스 정의를 위한 GUI

4.3 시스템 모델링

그래픽 이미지를 이용한 시스템 모델링 작업은 독립적으로 이루어질 수 있다. 시뮬레이터와 통신을 위해서 앞의 인터페이스 정의에서 지정한 프로토콜과 정확하게 일치되도록 이미지 컴포넌트의 특성을 정의해야 한다.

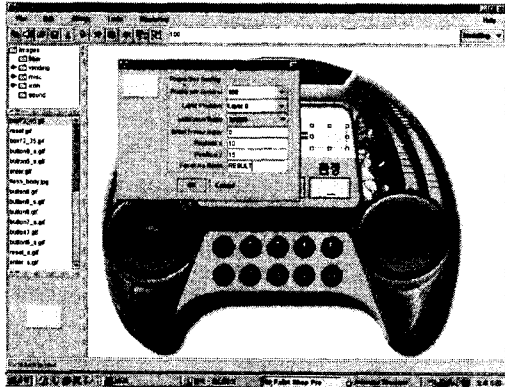


그림 7. 시스템 모델링

4.4 인터페이스 생성

사용자가 aif 파일에 시스템의 인터페이스를 간단히 기술하면 InterfaceGen 프로그램이 자동으로 인터페이스 프로그램을 생성해주도록 되어 있다. 자동 생성되는 인터페이스 프로그램은 별도의 파일로 생성되며 응용프로그램 개발자가 세부 내용을 파악할 필요가 없다. 응용 프로그래머는 인터페이스에 정의되어 있는 API를 이용해서 해당 부분만 코딩하면 된다.

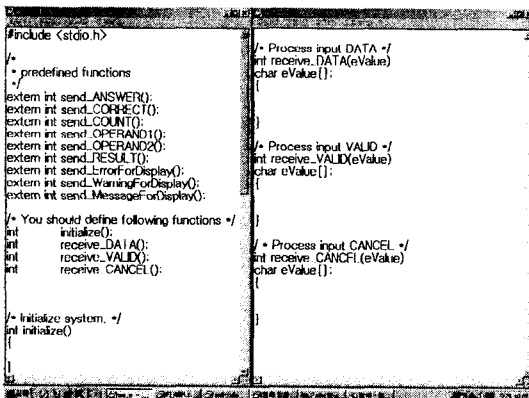


그림 8. 인터페이스 생성

4.5 시뮬레이터 작성

Client 프로그램인 java로 된 Animator 프로그램은 사용자로부터 이벤트를 받거나 외부 프로그램으로부터

이벤트를 받아서 애니메이션만 수행하고, 실제 시스템의 시뮬레이션은 server와 맞물려 있는 응용 프로그램이 담당한다. 응용프로그램에 대한 인터페이스는 자동으로 생성되므로 사용자는 인터페이스에 정의되어 있는 API를 이용해서 시뮬레이션 부분만 코딩하면 된다. 사용자가 직접 코딩해야 할 시뮬레이터 엔진에 해당하는 응용프로그램의 일부가 그림 9에 나와있다.

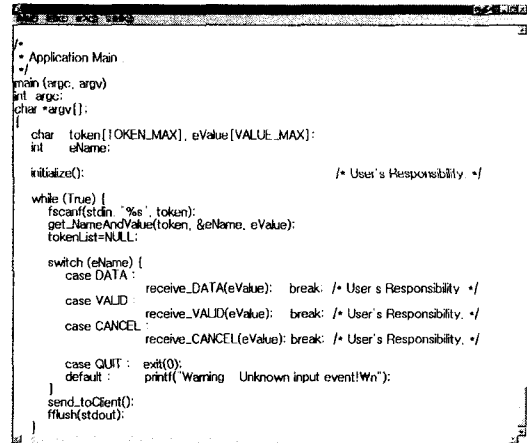


그림 9. 시뮬레이터 엔진 프로그램

V. 결론

시스템의 검증에 애니메이션 기법을 이용할 경우 시스템 개발에 참여하고 있는 모든 개발자들이 정확하게 목표 시스템의 기능을 파악할 수 있고, 시스템의 병목 현상이나 발생 가능성이 있는 오류를 미리 점검해볼 수 있다. 또한 애니메이션하는 프로그램이 java로 구현되어 있기 때문에 다양한 컴퓨터 기종에서 수행이 가능하다. 이 프로그램은 독창성이 있으며, 대상 시스템을 C-언어로 기술해서 시뮬레이션하기 때문에 애니메이션 속도가 실시간에 가깝게 이루어지는 장점이 있다.

본 논문은 시스템 검증을 시뮬레이터와 애니메이터를 연동시켜서 모델링하고 검증하고자하는 연구 분야에 기초 자료로 도움을 줄 수 있을 것으로 사료된다.

참고문헌

- [1] David R C Hill,, "Object-Oriented Analysis and Simulation" Addison-Wesley, pp.137-139, 1996.
- [2] 박인학 , "LODECAP ASIC Front-End 설계용 CAD 시스템", 시그마프레스, pp.93-94, 2000.4.