□ 특별기고 I □

CGI 적합성 시험전략

한국표준과학연구원 맹승렬*·박동순**

목 차

I. 서 론

적합성 시험의 목적은 표준 규격의 구현 시스템이 해당 표준이 규정하는 내용과 일치하는가를 테스트하여 이의 여부를 사용자나 생산자에게 알려줌으로서 구현 시스템에 대한 판단을 돕는데 있다. 미국내 적합성 시험의 현황은 프로그래밍 언어, 데이타 베이스 SQL, 그래픽 시스템, GOSIP, POSIX, 컴퓨터 보안에 대해 수개내지 수십개 회사의 제품이 적합성 시험통과 제품으로 등록되어 있다[10].

컴퓨터 그래픽 인터페이스(CGI)는 장치 독립적 방법으로 클라이언트 프로그램과 가상 장치 사이의 그래픽 정보를 교환하기 위한 수단을 제공하는 국제 표준이며 GKS, CGM, PHIGS 등과 함께 잘 알려진 그래픽 표준이다. CGI에 대한 국제 표준화 작업은 1989년 작업 초안(Working Draft)이 만들어진 이후 CGI 내용중 일부가 국제 표준(IS9636)으로 완성되었다. CGI 표준의 내용은 기능 정의(Functional Specification)와 언어 결합(Language Binding) 및 데이타 엔코딩 (Data Encoding)으로 구성되며, 기능 정의 부분이 IS 9636이 되었다. 그래픽 표준 GKS, CGM, CGI, PHIGS의 적합성 시험 소프트웨어는 미국 및 유럽에서 개발되었으며, PHIGS를 제외한 그래픽 표준 구현 시스템에 대해 시험 서비스를 실시하고 있다. 그래픽 표준들은 각기 그 적용 범위와 목적이 다르므로 개발된 적합성 시험 소프트웨어들은 동일한 시험 방법을 사용하지 않는다. 다만 그래픽 표준 적합성 시험 소프트웨어 개발의 일반적 원칙 및 요구사항이 국제 규격으로 규정되어 있다[9]. 이외에도 그래픽 표준 적합성 시험 소프트웨어의 요구사항에 대해서는 여러 논문이 있으며[1,2,3,6,9]는, 각각 CGM, CGI, PHIGS의 전체적인 적합성 시험절차를 제안하고 있다. 특히, 1988년 CGI 워크샤은 CGI의 적합성 방법에 대한 여러편의 논문이 발표되었다. [5]는 CGI 출력 테스트 방법을 제안하였다. [4]는 CGI 세그먼트의 테스팅 전략을, [7]은 CGI의 래스터 함수의 테스팅 전략을 제안하였다.

본 논문에서는 지금까지 제안된 적합성 시험 요소 이외에 적합성 시험에서 꼭 필요한 요소로

* 정회원
** 종신회원

파이프라인 테스트와 오류처리 테스트의 필요성을 설명하고, 테스트 전략을 제시하고자 한다. 본 논문에서 제안하는 적합성 시험 전략의 특징은 CGI 동작특성을 테스트 하는데 있다. 지금까지의 CGI 테스트가 정적인 테스트 였다면, 본 논문에서 제안하는 전략은 CGI의 동적인 특성을 테스트 한다.

II. 그래픽 적합성 시험의 원리

적합성 시험의 개념은 구현된 시스템이 오류를 포함하고 있지 않다는 것을 보장하는 것이 아니라 특정 테스트 항목에 대해 오류가 없다는 것을 밝히는 것이다. 즉, 주어진 조건들에 대해 테스트 항목의 검사결과를 관찰한다.

컴퓨터 그래픽 인터페이스 표준을 논리 시스템의 공리들의 집합으로 생각할수 있다. 공리의 형태는 다음과 같다-"모든 X에 대해 만일 X가 표준을 따르는 구현이라면, X는 다음과 같이 동작한다". 즉 표준을 따르는 구현이 동작해야 하는 기능을 정의한다. 표준 규격에 따라 구현 되었다고 주장하는 시스템 P를 테스트할 때 추가적인 공리, "P는 표준 규격을 따르는 구현이다"가 필요하다. 이들 공리로부터 P가 갖는 동작 특성에 대해 많은 정보를 추론할 수 있다. 만일 P의 동작 기능이 이론적으로 정의된 기능과 다르다면 P가 따르는 공리는 거짓이라는 결론을 얻을 수 있다. 비록 이러한 적합성 시험 원리가 이론적으로는 쉽게 생각될 수 있다. 그러나 실제 적용에서는 많은 문제점을 갖고 있다.

첫째 표준 규격은 논리적 공리들로 이루져 있지 않다-표준 규격은 문장 형태로 서술되어 있다. 둘째 아주 간단한 결론을 유도하기 위해 표준 규격중 직접적으로 관계되지 않는 규정 내용으로부터의 공리들이 필요하다. 셋째 논리 추론에서 사용하는 추론 체계를 직접 프로그램으로 코딩 한다는 것은 매우 어렵다. 그렇지만 논리 추론모델은 실제 시스템을 구조화 하는데 좋은 모델을 제공한다.

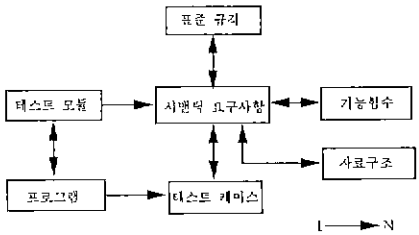
표준규격의 공리에 해당하는 "시맨틱 요구사항(Semantic Requirement)"과 논리 시스템의 결론에 해당하는 "테스트 케이스(Test Case)"를 정의한다. 적합성 시험 도구는 여러개의 모듈로 구성되며 각각의 모듈들은 SR의 집합을 포함한다. 이들 SR은 표준 규격으로부터 만들어진 공리이다.

단일 SR로부터 구현 시스템이 갖추어야할 동작 특성을 유추하는 것은 불가능하다. 보통은 여러 SR로부터 직접 테스트할 수 있는 결론이 얻어진다. 이와같은 결론은 TC의 기본이 된다. 적합성 시험 도구는 SR의 집합 뿐 아니라 각각의 프로그램에 TC의 집합을 포함하도록 구성된다. 이들 TC들은 적합성 시험 도구의 수행 가능한 코드이다. 중요한 것은 이들 TC를 어떻게 다시 해당 SR과 연관시키는가 하는 것이다. 각 TC는 SR의 집합과 매핑된다. 일반적으로 SR과 TC 사이에는 N : N 관계가 존재한다. 즉 하나의 TC는 보통 여러개의 SR에 종속되며, 각 SR은 여러 TC에 사용될 수 있다 . 표준 구현이 TC를 통과 한다고 해서 그 구현이 정말로 표준 규격을 정확히 따르고 있다고 주장할 수 없다. 즉 테스트 되지 않은 부분에서 표준 규격과 일치하지 않을 수 있다. 그러나 적당한 TC를 통과하지 못한 구현은 곧 표준 규격과 일치하지 않음을 의미한다.

지금까지 살펴본 적합성 시험의 논리적 특성과 그래픽 시스템의 입출력 구조로부터 적합성 시험 소프트웨어에 포함될 객체들과 이들의 관계를 정의할 수 있다. 이는 곧 적합성 시험 도구의 개념적 모델이 된다. 적합성 시험 도구에 사용될 객체들은 다음과 같다.

- ○ 그래픽 함수
- ○ 그래픽 소프트웨어의 자료구조
- ○ 표준 규격으로 부터 얻어지는 SR
- ○ 그래픽 표준 규격 TC

테스트 도구의 사용자는 이들 객체간의 관계를 정확히 알 필요가 있다. 특히 적합성 시험은 특정 검사 항목과 표준 규격과의 관계에 대해 정보를 줄 수 있어야 한다. 적합성 시험 도구를 설계함에 있어 두가지 원칙이 고려되어야 한다. 테스트는 논리적으로 타당해야 하며, 테스트과정에서 생성되는 결과는 표준 규격을 기반으로 얻어진 다는 것을 보여주어야 한다. 데이타베이스 설계 기법이 이러한 문제를 설명할 수 있는 좋은 테



(그림 1) 적합성 시험 오브젝트들 간의 관계

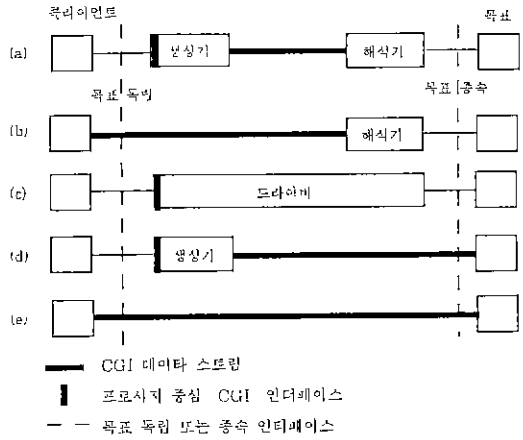
크닉이다. 적합성 시험에서 사용하게 되는 객체들간의 관계를 데이타 베이스 스키마 (그림 1)로 표현한다.

SR은 논리적 데이타베이스 시스템에서 중심 역할을 수행한다. 즉 그래픽 함수의 동작과 데이타에 대해 설명한다. 따라서 SR은 관련된 객체들의 참조 포인트가 된다. 특히, 각 SR은 관련된 그래픽 함수, 자료구조, 표준 규격의 참고사항 등의 자료를 포함한다. 그래픽 함수와 자료구조, 참고사항에 대한 자료들이 일정한 형태로 구성되므로 상호 참조 테이블을 만들수 있다.

III. CGI 적합성 시험요소의 선정

3.1. CGI 기능

컴퓨터 그래픽 인터페이스의 목적은 클라이언트 프로그램과 그래픽 가상 장치 사이의 그래픽 정보를 서술하고, 이들 간에 그래픽 정보를 장치 독립적인 방법으로 교환하기 위한 표준화된 인터페이스를 제공하는데 있다. 이 인터페이스는 소프트웨어 대 소프트웨어(네트워크 환경에서 사용되는 데이타 스트림 엔코딩, 또는 소프트웨어 패키지에 대한 언어 결합) 또는 소프트웨어 대 하드웨어(디바이스 프로토콜로 그래픽 인터페이스를 엑세스하는 디바이스의 경우 데이타 스트림 엔코딩)인터페이스로 구현된다. 그래픽 참조 모델은 전체 그래픽 시스템에서 CGI의 개념적 위치를 보여주며, 클라이언트 프로그램과 가상장치 사이에 CGI가 구현될 수 있는 형태를 도시하고 있다[8]. (그림 2)는 CGI 참조 모델에서 인용한 CGI 구현 형태를 도시한 것이다.



(그림 2) 그래픽 인터페이스 참조 모델

(그림 2-(a))는 가장 기본적인 그래픽 인터페이스 모델이다. CGI 해석기로 전송될 CGI 데이타 스트림을 만들기 위해 CGI 생성기 서브루틴 라이브러리의 프로시져 결합을 사용한다. 클라이언트와 생성기 사이의 인터페이스는 장치 독립적이다. 목표시스템 (Target)은 장치라는 말보다 일반적이기 때문에 목표 독립적이라는 단어로 표기되어 있다.

(그림 2)의 그래픽 시스템 배열형태 (b), (c), (d) 그리고 (e)는 이 기본형태로부터 얻어지는 그래픽 시스템 형태이다. 그림 (a), (c) 그리고 (d)는 프로시져 결합 CGI 인터페이스 이다. 그러나 CGI 데이타 스트림이 존재한다는 것은 매우 중요하다. CGI 생성기 구현에서 클라이언트는 CGI 데이타 스트림이 흐르는 입/출력 연결을 최적화하기 위해 생성기의 엔코딩 방법을 제어할 수 있다. 예를들면, 어떤 그래픽 응용 시스템은 32비트 가상 장치 좌표계를 사용하는 반면 다른 그래픽 시스템은 16비트 정수형 가상 장치 좌표계를 사용할 수 있다.

(그림 2-(b))에서 CGI 데이타 스트림 생성기의 역할이 클라이언트 프로그램에 흡수되어 있다. 이 경우, 클라이언트는 CGI 프로시져 결합을 사용할 필요가 없이 CGI 데이타 스트림 자체를 생성한다. 목표 시스템에 독립적인 인터페이스는 표준 데이타 스트림 엔코딩이다.

(그림 2-(c))에서 생성기와 해석기가 드라이브

형태로 결합되어 있다. 이 경우 CGI 데이타 스트림은 더이상 필요없다. 서브루틴 라이브러리는 목표시스템과 직접 통신할 수 있다. (d)와 (e)에서 CGI 생성기는 목표 시스템과 통합되어 있다. 특히 CGI 가상 장치의 구현은 목표 시스템이 될 수 있다. 그림 (e)는 자체가 CGI 해석기인 목표 시스템으로 직접 CGI 데이타 스트림을 전송하는 클라이언트를 도시한 것이다.

3.2 CGI 구성과 동작 원리

CGI 기능 함수는 가상 장치를 제어하는데 필요한 제어 함수, 그래픽 출력을 위한 그래픽 프리미티브 함수, 단위 그래픽 출력을 복합 단위로 처리하기 위한 세그먼트 함수, 그래픽 입력을 위한 입력 함수, 래스터 그래픽 데이터 처리를 위한 래스터 함수로 이루어지며, 이들을 엑세스하기 위한 절차는 로컬 환경을 위한 언어결합과 네트워크 환경을 위한 엔코딩 방법으로 나눌 수 있다.

한편 CGI 구성 요소들로 이루어진 CGI 구현 시스템의 동작은 파이프라인 모델로 설명될 수 있다. CGI 그래픽 오브젝트 파이프라인은 CGI 가상 장치의 추상적인 동작을 설명하기 위한 개념적인 모델이다. 이것은 CGI 가상 장치의 구현 방법을 제시하는데 목적을 둔 것은 아니다. 파이프라인은 4가지 기본 엔터티를 포함한다.

- 파이프 : 그래픽 오브젝트가 지나가는 통로, 세그먼트 스토리지와 드로잉 표면(Drawing Surface)은 이 통로에 위치한다.
- 오퍼레이션 : 그래픽 오브젝트가 파이프를 따라 지나가는 동안 그래픽 오브젝트에 적용될 수 있는 프로세스
- 상태리스트 정보 : 오퍼레이션 수행과 관련한 정보
- 그래픽 오브젝트 주석 : 파이프라인의 특정 지점에서 그래픽 오브젝트가 표현되는 방법을 설명한다.

그래픽 오브젝트가 CGI 파이프라인을 따라 흐를 때 가해지는 오퍼레이션에 따라 CGI 상태가 계속적으로 변화한다. 연속적인 CGI 상태 변화는 상태 천이모델로 설명할 수 있다. 한 시점에서 CGI 상태는 상태 변수의 집합으로 정의된다. 상태 변수의 집합으로 표현되는 CGI의 상태는 여기에 가해지는 오퍼레이션에 의해 상태변수의 값이 변하고, 이는 CGI의 또다른 상태를 나타낸다. 상태 천이를 야기하는 CGI 오퍼레이션은 다음 중 하나이다.

- CGI가 정의하는 기능 함수
- 오퍼레이터의 시스템 작동

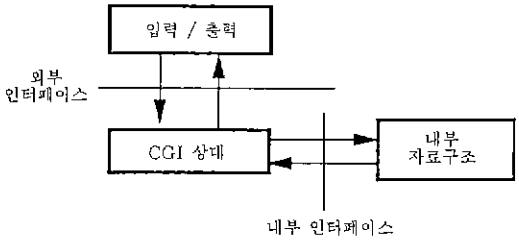
따라서 CGI 파이프라인은 연속적인 상태 천이로 생각할 수 있다. 그러나 CGI 파이프라인은 특정 지역에서 특정함수 사용를 금지하고 있다. 즉 현제 CGI 상태에서 수행할 수 없는 오퍼레이션이 정해져 있다. 만일 특정 CGI 상태에서 허용되지 않는 함수를 사용하게 되면 CGI 구현 시스템은 오류 처리 메카니즘의 동작을 보장해야 한다. 여기서 CGI 적합성 시험의 테스팅 포인트를 선정해 보자.

3.3 CGI 적합성 시험 요소의 선정

CGI 구현 시스템의 무엇을 테스트해야 적합성을 인정할 수 있는 가가 적합성 시험 소프트웨어 설계에서 우선적으로 고려되어야 한다. 개념적으로, 모든 가능한 함수의 오퍼레이션에 대해 CGI 파이프라인 상의 상태 변화가 타당하면, 그 CGI 구현 시스템은 적합하다고 할 수 있다. 또한 부당한 CGI 함수 오퍼레이션에 대해서는 CGI 오류처리 메카니즘에 따라 해당 유형의 오류처리 결과를 생성해야만 적합성을 인정할 수 있다.

따라서 CGI 적합성은 먼저 CGI 파이프라인 오퍼레이션의 적정성과 부당한 CGI 함수 오퍼레이션에 대한 CGI 구현 시스템의 반응 결과에 의해 판단할 수 있다. 한편 CGI는 다양한 입력 방법과 출력 프리미티브를 정의하므로 정의된 입력 기능과 출력 프리미티브가 모두 구현 되었는 가를 테스트 해야만 된다. 이는 CGI 구현 시스템의 내부 테스팅 요소와 외부 테스팅 요소를 의미한다. 이들의 관계를 그림으로 도시하면 (그림 3)와 같다.

(그림 3)의 외부 인터페이스는 오퍼레이타의



(그림 3) CGI의 내부/외부 테스팅 요소

개입이 필요한 요소를 나타내며, 내부 인터페이스는 오퍼레이타의 개입이 불필요하다는 것을 나타낸다.

일반적인 적합성 시험은 두가지 테스팅 방법을 사용한다. 하나는 오퍼레이터의 개입이 필요 없는 자동검사이고, 다른 하나는 오퍼레이터의 개입이 필요한 수동검사이다. 다음은 자동검사와 수동검사에 포함되는 시험요소이다.

- ○ 자동검사 : CGI 파이프라인 상의 상태 천이가 적절이 이루어지고 있는가를 테스트하는 파이프라인 테스트. 부당한 함수의 수행으로 야기되는 오류 처리 결과의 타당성을 검사하는 오류 테스트.
- ○ 수동검사 : 오퍼레이타의 개입을 필요로 하는 검사로서 오퍼레이타 직능검사에 의해 적합성 판정이 이루어지는 검사. 출력 프리미티브 종류 및 형태에 대한 테스트

여기서 우리의 관심인 파이프라인 테스트와 오류 테스트는 자동검사에 해당된다. 자동검사에서 수행하게 되는 파이프라인 테스트와 오류 테스트의 테스트 범위는 다음과 같이 정의된다.

- ○ 파이프라인 테스트 : 하나의 오퍼레이션 결과에 의해 변화된 CGI 상태의 적합성을 테스트함.
- ○ 오류 테스트 : 파이프라인상의 특정 CGI 상태에서 부당한 오퍼레이션을 수행했을 때 오류 처리 결과의 적합성을 테스트함.

파이프라인 테스트는 이상적으로 CGI의 모든 함수의 수행에 대해 CGI 상태 천이를 테스트해야 한다. 이 경우 최소한 IS9636의 5개 CGI 함수 그룹의 조합가능한 숫자 만큼의 테스트 케이스가 존재한다. 그러나 대부분의 경우 CGI 함수는 하나의 그룹내 함수들이 여러개 수행된 후 다른 그룹의 함수를 사용하게 되므로 이상적으로 조합가능한 테스트 케이스는 필요하지 않다. 또한 CGI는 파이프라인 상의 특정 CGI 상태에서 허용되지 않는 함수를 포함한다. 예를들면 CGI 함수 B가 수행되기 위해서는 함수 A가 선행되어야 한다.

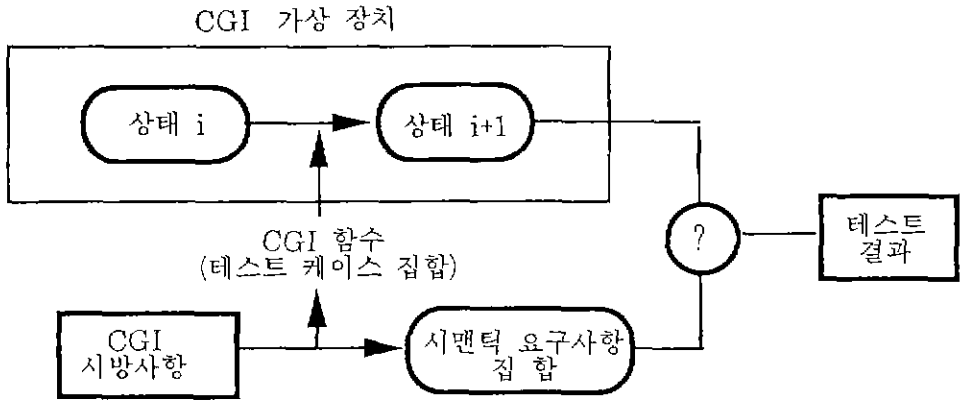
테스트 케이스 결정에 이러한 제약 조건을 추가하면 실질적으로 필요한 파이프라인 테스트의 수는 상당히 감소한다. 파이프라인 테스트에 사용하는 CGI 함수의 적용 순서는 적합성 시험의 효율성과 관련된다. 간단히 생각하면 CGI 함수를 임의의 순서로 파이프라인 테스트에 적용하는 경우와 IS9636의 각 기능별 분류 기준에 따라 함수의 적용 순서를 결정하는 방법이 있을 수 있다. 전자의 경우 파이프라인 테스트의 테스트 케이스 결정이 쉽다는 측면이 있는 반면, 적합성 시험 프로그램의 모듈성이 떨어지며, 무엇보다 적합성 시험의 체계성이 없다. 반면에 후자의 경우는 전자의 단점을 보완할 수 있다. 따라서 파이프라인 테스트의 함수 테스트는 IS9636에서 구분한 기능별로 테스트 순서를 정하는 것이 바람직하다. 이 경우도 모든 조합가능한 테스트 케이스가 가능하다.

오류 테스트는 부당한 함수 사용에 대해 CGI 구현 시스템의 반응을 테스트해야 한다. 따라서 정상적인 CGI 파이프라인 상에서 계획된 부당한 CGI 함수를 사용하고, 이의 반응 결과를 기대값과 비교한다. 따라서 오류 테스트의 테스트 포인트는 파이프라인 테스트와 같다.

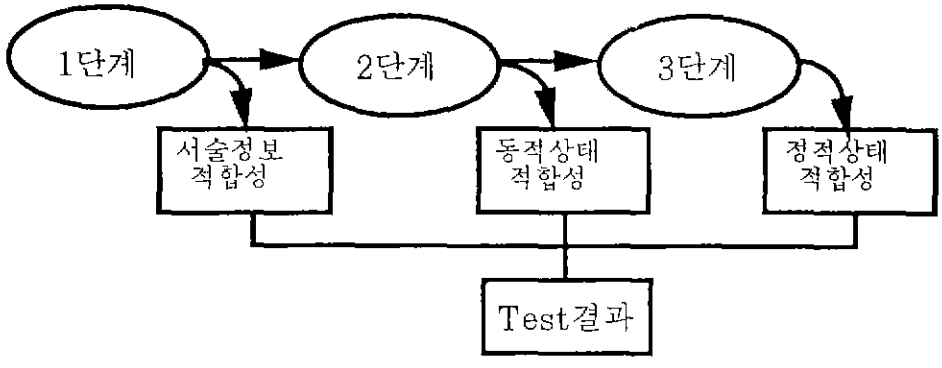
IV. 파이프라인 및 오류처리 테스트 전략

4.1. 파이프라인 테스트 전략

파이프라인 테스트에서는 CGI 함수 호출에 의해 연속적으로 변하는 CGI 가상 장치의 상태를



(그림 4) 파이프라인 테스트 기본 개념



(그림 5) 파이프라인 테스트 흐름도

테스트한다. CGI 가상 장치의 상태는 상태 변수들의 집합으로 표현되며, 이것은 CGI 내부 자료구조에 의해 구현되므로 파이프라인 테스트의 궁극적인 목표는 내부 자료구조 데이타 값을 검사하는 일이다.

CGI 내부 자료구조는 각 기능의 서술 정보 테이블과 상태 리스트로 구성된다. 따라서 파이프라인 테스트는 서술 정보 테이블과 상태 리스트 값의 테스트로 요약될 수 있다. 그런데 파이프라인 테스트의 기본 개념은 CGI 함수 호출에 의해 상태 변화가 올바르게 이루어 졌는가를 검사하는 것이므로 CGI 함수 호출후 상태 리스트 테이블의 값이 규격에서 요구하는 값으로 대치되었는가를 검사하면 된다. 이를 그림으로 표현하면 (그림 4)와 같다.

(그림 4)에서 박스 부분은 CGI 가상 장치를 의미하며, 파이프라인 테스트를 목적으로 구현 특성과 상태 변화로 CGI를 요약 표현한다. CGI 가상 장치의 상태 변화는 CGI 함수 호출에 의해 이루어지므로 CGI 상태 변화를 야기하는 CGI 함수와 파라미터 값의 선정이 테스트 케이스 생성에 해당한다. 테스트 케이스 생성은 표준 규격으로부터 각 함수의 파라미터 값과 그 함수 결과에 대한 분석에 의해 이루어지는데 분석 결과는 사용되는 목적에 따라 CGI 가상장치의 테스트 케이스와 CGI 가상 장치의 요구사항으로 사용되게 된다.

테스트 결과는 CGI 함수 호출 결과와 CGI 가상 장치 요구사항의 비교에 의해 만들어 지는데, 이것의 CGI 구현 시스템의 적합성 판정 기준이 된다.

CGI의 내부 자료구조를 구성하는 서술 정보

테이블과 상태 리스트 테이블은 각각 그 특성을 갖고 있다. 서술 정보 테이블은 CGI 가상 장치의 구현 특성에 관한 정보를 포함하며, 외적인 변환이 없는 한 일정하다. 반면에 상태 리스트는 CGI 가상 장치의 구동 상태에 관한 상세한 정보를 갖고 있다. 상태 리스트는 CGI 함수 호출에 의해 일시적으로 만들어지는 동적인 상태 리스트와 CGI 구현시 고정되는 정적인 상태 리스트로 구분된다.

제 1단계 테스트는 서술 정보 테이블의 적합성 검사이다. 이 결과는 CGI 가상 장치의 각 기능이 올바로 구현 되었는가를 나타낸다. 2단계와 3단계 테스트 결과는 CGI 가상 장치의 동작 상태에 대한 적합성을 의미한다. CGI 함수 호출에 대해 CGI 가상 장치가 적합한 행동을 보여주고 있다는 것을 말한다.

4.2. 오류처리 테스트 전략

오류 처리 테스트는 부당한 CGI 함수 호출에 대해 CGI 구현 시스템의 반응 동작을 테스트 하므로, CGI 구현 시스템의 오류 처리 메카니즘이 동작하도록 해야 한다. IS9636의 9가지 오류 클라스에 해당하는 테스트 케이스를 만들고 각각의 함수를 호출한다. 오류 처리 테스트는 기본적으로 파이프라인 테스트와 유사하다. 그러나 오류 처리 테스트는 반드시 함수 호출후 테스트 결과가 발생하므로 파이프라인 테스트에서 서술 정보 테스트와 정적인 상태 리스트 테스트 시점에서의 오류 처리 테스트는 불가능하다. 반면 파이프라인 테스트의 동적인 상태 리스트 테스트는 연속적인 CGI 함수 호출을 통해 CGI 상태 천이를 테스트 하므로, 오류 처리 테스트 시점은 동적인 상태 리스트 테스트 시점이 된다.

오류 처리 테스트는 적합성 시험 소프트웨어를 구현할 때 별개의 테스트 모듈로 구현하는 것이 효율적이므로, 오류 처리 테스트 전략에서는 오류 클라스의 테스트 순서 결정이 필요하다. IS 9636의 오류 클라스는 9가지로 분류된다. 이들중 클라스 1부터 3은 CGI 함수 호출에 관련된 클라스이다. 이 클라스는 입력 파라미터와 출력 파라미터 값에 관한 오류, 그리고 호출된 함수의 수행 가능 여부에 대한 오류이다. 클라스 4에서 5는 함수의 구현 상태에 대한 오류이다. 호출한 함수가 구현된 기능인가, CGI 상태가 호출된 함수를 수행할 수 있는가에 관한 오류이다. 클라스 6, 7은 CGI 함수 수행과 관련한 시스템 환경에 관한 오류이다. 오류 클라스 8, 9는 비표준화된 함수에 관한 오류 클라스이다.

V. 결 론

CGI 적합성 시험 소프트웨어 개발을 위해 맨처음 해야할 일은 CGI 구성요소중에서 무엇을(what), 어떠한 방법(How)으로 테스트할 것인가 선정하는 작업이다. 이것은 CGI 뿐만 아니라 다른 그래픽 표준, CGM, CGI, GKS, PHIGS의 적합성 시험 소프트웨어 개발에도 마찬가지이며, 나아가 다른 분야의 표준 규격에 대한 적합성 시험 소프트웨어 개발에도 적용된다. CGI와 다른 그래픽 표준의 적합성 시험에 관해 여러편의 논문이 발표되었다. 이들은 그래픽 표준의 적합성을 판정하기 위해 필요한 시험 요소와 전략을 제시하고 있다.

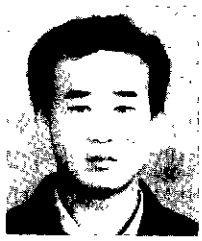
본 논문에서는 CGI의 적합성 판정 기준이 되는 시험 요소를 CGI 내부적 요소와 외부적 요소로 나누고, 내부적 시험 요소로 CGI 동작 상태인 그래픽 오브젝트 파이프라인의 적합성 시험의 필요성과 시험 전략에 대하여 논하였다. 이밖에도 CGI의 완전한 적합성 판정을 위해서는 다른 시험 요소가 있을 수 있다. 파이프라인 테스트는 CGI 구현 특성과 내부 동작을 테스트하는데 목적을 두고 있다.

그러나 본 논문에서 제시한 CGI 적합성 시험 전략은 파이프라인 테스트의 기본 개념에 대해서만 제시하고 있어 이를 실제 구현하기 위해서는 IS 9636의 각 기능 함수에 대한 세부 시험 전략 즉, 시맨틱 요구사항과 테스트 케이스가 결정되어야 한다.

참고문헌

1 A. Mumford and J. Pink, "Proposing a Test Model for the Comptuer Graphics Metafile", Com-

puter Standards and Interfaces **8**, 1988, 189~197.
2 B. Kirsch, C pfluger and C. Egelhaaf, "Conformance Testing for Computer Graphics Standards", Computer Standards & Interfaces **12**, 1991, 35~42.
3 R. Ziegler, M. Gobel, "Conformance Testing of Implementations of Graphics Standards", Conference Proceedings on ASP-90, 1990. 9. Germany.
4 C. Egelhaaf, "Testing of Segments", Eurographics Workshop on Applications Test Methods for CGI", 1988. 5, Heppenheim, France.
5 M. R. Gomes, "Testing CGI Output", Europraphic Workshop on Application & Test Methods for CGI, 1988. 5, Heppenheim, Frnace.
6 G. Guy *et al*, "Conformance Testing at the Computer Graphics Interface", Europraphic Workshop on Applications & Test Methods for CGI, 1988. 5, Heppenheim, France.
7 M. Gobel and R.Ziegler, "Strategy for Raster Function Testing in CGI Implementation", Eurographic Workshop on Applications & Test Methods for CGI, 1988. 5. Heppenheim, France.
8 Information Processing System-Computer Graphic Interfacing Techniques for Dialogues with Graphical Devices, IS9636/1~6, 1991.
9 Information Processing System - Conformance Testing of Implementations of Graphics Standards. IS10641, 1992.
10 J. B. Kailey, "Validated Products List", NISTIR 5103, NIT, Gaithersberg, MD. 1990.



맹 승 렬

1984 충남대학교 계산통계학과 학사
1986 충남대학교 대학원 계산통계학과 석사
1992 과학기술원 박사과정
1984 ~현재 한국표준과학연구원 선임연구원
관심 분야 : 컴퓨터 그래픽스

박 동 순

1968 청주대학교 경제학과 학사
1976 고려대학교 경영대학원 석사
1984 청주대학교 경영학과 박사
1971 ~1978 KIST/SEC 선임연구원
1979 ~1984 KNFC, KAERI 전산, 기술지원 실장
1986 ~1987 미 국립표준국 초청연구원
1984 ~현재 한국표준과학연구원 전산센타 실장
관심분야 : 한국어 정보처리, 정보기술표준화