컴포넌트 기반 프로그램 분석을 위한 시퀀스 다이어그램 자동 생성 방법

임일명, 임보희, 강신일, 최병호
전자부품연구원
ilmyeongim@keti.re.kr, bohee80@keti.re.kr, sinil@keti.re.kr, bochoi@keti.re.kr,

A method for automatically creating operation sequence diagram of component based program

Il-Myeong Im, Bo-hee Im, Sin-il Kang, Byeon-Ho Choi
Korea Electronics Technology Institute

요 약

컴포넌트 기반 프로그램은 동시에 여러 컴포넌트간의 상호 협력에 의해서 동작한다. 여러 개의 컴포넌트 기반 프로그램을 기존의 순차적 분석 방법을 이용하여 컴포넌트의 동작을 분석해야 하기 때문에 시간이 많이 소요되고 매우 불편하다. 더욱이 컴포넌트의 수가 많아지면서 각 단계마다 각각의 컴포넌트의 동작을 모두 추적한다는 것은 매우 어렵다. 본 논문에서는 컴포넌트 기반으로 설계된 프로그램의 동작을 나타내는 시퀀스 다이어그램을 자동으로 생성할 수 있는 방법을 제안한다. 컴포넌트 기반 프로그램의 UML 시퀀스 다이어그램을 자동으로 생성하여 시각적으로 보여주어 개발의 편의성과 효율성을 증가시킨다.

1. 서론

소프트웨어의 구조가 갈수록 복잡해지고, 컴포넌트 기반의 소프트웨어 개발이 중요해졌다. 또한, 최근의 CPU 들은 대부분 멀티 코어 구조로 개발 된다. 프로그램도 이를 활용하여 개발되고 있어, 컴포넌트 기반의 소프트웨어 개발이 증가하고 있다.

컴포넌트 기반 프로그램은 동시에 여러 컴포넌트간의 상호 협력에 의해서 동작한다. 기존에는 이러한 여러 개의 컴포넌트 기반 프로그램을 순차적 분석 방법을 이용하여 각 컴포넌트의 동작을 분석해야 하기 때문에 매우 불편하고 시간이 많이 소요된다. 더욱이 컴포넌트의 수가 많아지면 분석은 더욱 어려워진다. 이 경우, 분석 대상 컴포넌트 이외의 다른 여러 컴포넌트의 동작을 보류하고 분석을 수행하는 방법이 있지만, 일부 컴포넌트의 동작을 보류 시켰기 때문에 분석 환경이 프로그램의 실제 동작 상황과 달라져 분석과정 중에 문제가 동일하게 재현되지 않을 수 있다는 문제점이 있다.

앞서 말한 바와 같이 기존의 분석 방법으로 컴포넌트 기반 프로그램을 분석하기에는 매우 시간이 많이 소요되고 어렵기 때문에 새로운 분석 방법이 필요하다. 본 논문에서 제안한 컴포넌트 기반으로 설계된 프로그램 시퀀스 다이어그램을 사용하게 되면, 컴포넌트 기반 프로그램의 UML[1] 시퀀스 다이어그램을 자동으로 생성하여 시각적으로 보여주어 개발의 편의성과 효율 증가된다.

본 논문은 UML 시퀀스 다이어그램 자동생성을 통하여 컴포넌트 기반 소프트를 분석하는 편리하고 효율적인 방법을 새로 제시한다. 한 예로, 제안한 방법은 OpenMAX IL[2]이나 DirectShow[3] 등의 미디어 프레임워크를 이용한 응용프로그램에서 사용 가능하다. OpenMAX IL 을 사용하여 제작된 동영상 재생 프로그램에 적용하여 결과를 보여준다.

2. 제안 방안

본 논문에서는 동작을 분석할 컴포넌트를 등록하는 기능, 컴포넌트의 동작을 저장하는 기능을 제공하고, 저장된 컴포넌트의 정보와 동작로그를 이용하여 시퀀스 다이어그램을 생성하는 라이브러리를 설계했다.

분석할 응용프로그램은 라이브러리를 초기화 하고, 응용프로그램의 각 컴포넌트는 라이브러리가 제공하는 컴포넌트 등록 함수를 호출하여 컴포넌트 정보를 등록하고, 컴포넌트 사이에 동작이 일어날 때 마다 동작 로그 함수를 이용하여 컴포넌트의 상태전의 컴포넌트간의 통신 기록을 저장한다. 이 저장된 정보를 바탕으로 UML 시퀀스 다이어그램을 자동으로 생성한다.

시스템은 그림 1 은 제안한 라이브러리를 사용한 컴포넌트 기반 응용프로그램의 구조를 설명한다. 응용프로그램은 자신과 컴포넌트 정보 저장기, 컴포넌트 로그 저장기, 시퀀스 다이어그램 생성기를 갖는다. 컴포넌트 정보 저장기는 프로그램에 소속되어있는 컴포넌트 등록하고 이름과 컴포넌트 번호를 저장한다. 컴포넌트 로그 저장기는 각 컴포넌트 사이의 동작 로그를 시간과 함께 저장한다.

시퀀스 다이어그램 생성기는 등록된 컴포넌트 정보와 동작 로그를 기반으로 시퀀스 다이어그램을 생성한다. 컴포넌트 정보 저장기, 컴포넌트 로그 저장기, 시퀀스 다이어그램 생성기를 분석할 프로그램과 분리해서 분석할 프로그램이 비정상 동작

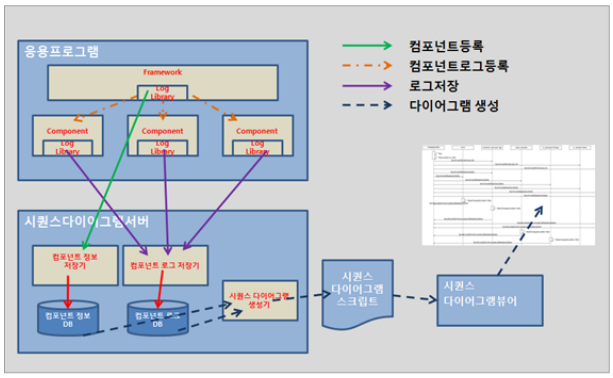

그림 1. 제안된 라이브러리를 사용한 응용프로그램 구조

하더라도 다이어그램을 생성할 수 있다.

프로그램은 컴포넌트를 생성하면서 컴포넌트를 컴포넌트 정보 저장기에 등록하고 컴포넌트 번호를 부여 받는다. 컴포넌트가 해제되면 컴포넌트 정보 저장기에서 제거한다. 컴포넌트가 다른 컴포넌트에게 동작을 요청하면, 컴포넌트 로그 저장기에 기능을 요청하는 컴포넌트번호와 요청 받는 컴포넌트 번호 요청의 종류를 저장한다. 컴포넌트 스스로가 상태를 변경하거나 컴포넌트 로그 저장기에 컴포넌트 번호와 변경된 상태 정보를 남긴다. 컴포넌트가 스스로 내부 기능을 사용하면 컴포넌트 로그 저장기에 컴포넌트 번호와 변경된 상태 정보를 남긴다.

UML 시퀀스 다이어그램 스크립트 생성기는 컴포넌트 정보 저장소에 저장된 정보를 이용해서 다이어그램에 표시할 컴포넌트 리스트를 만들고, 컴포넌트 로그 저장소에 저장된 컴포넌트 번호와 요청 및 응답 종류를 이용하여 다이어그램 스크립트를 생성한다. 다이어그램 스크립트에서는 응용프로그램과 컴포넌트의 이름이 번호와 함께 나열되고, 응용프로그램과 컴포넌트간의 통신이 기록이 시간 순서대로 나열된다. 통신 기록은 통신을 요청한 컴포넌트와 수신한 컴포넌트의 이름, 요청의 종류, 요청 시간 등이 표시된다. 스크립트는 오픈소스 프로그램인 UMLet 이 정의한 XML 형식으로 생성한다.

UML 시퀀스 다이어그램 스크립트를 오픈소스 프로그램인 UMLet[4]을 사용하여 표시할 수 있다.

3. 실험 결과

제안한 방안을 확인하기 위하여 라이브러리와 OpenMAX IL 을 사용한 응용프로그램을 개발했다. 응용프로그램은 라이브러리를 초기화 함수를 호출하여 컴포넌트 저장소와 로그 저장소를 초기화한다. 응용프로그램을 구성하는 OpenMAX IL 컴포넌트는 생성시에 라이브러리를 사용하여 자신의 정보를 등록하고 동작을 기록하도록 제작했다.

프로그램을 동작시킨 후에 그림 2 와 같은 시퀀스 다이어그램이 생성되었다. 생성된 시퀀스 다이어그램에는 컴포넌트의 종류와 각각의 통신과정 컴포넌트의 상태변이가 나타난다. 이 다이어그램으로 OpenMAX IL 응용프로그램의 내부 동작방식을 쉽게 이해하고, 개발 중에 발생하는 문제점을 쉽게 발견할 수 있었다.

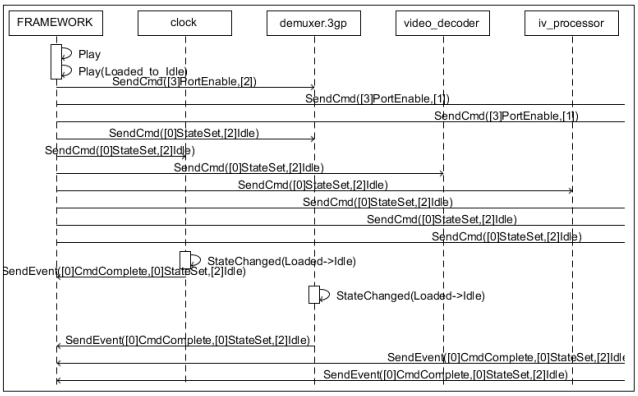

그림 2. 생성된 시퀀스 다이어그램

4. 결론

기존의 단계적 추적 방식의 디버깅 기법으로 컴포넌트 기반 응용프로그램의 동작 방식이나 문제점을 찾으려면 동시 여러 컴포넌트의 동작을 추적해야 하기 때문에 매우 불편하고 시간이 많이 소요된다. 본 논문에서는 컴포넌트의 UML 시퀀스 다이어그램을 자동으로 생성하여 시각적으로 보여주는 방안을 제시했다.

OpenMAX IL 을 사용한 응용프로그램을 제작하고 제안 방안을 활용하여 실험한 결과 UML 시퀀스 다이어그램이 생성되었고, 응용프로그램의 동작 방식과 문제점을 쉽게 파악할 수 있었다.

향후, 본 연구내용을 단일 시스템에서 동작하는 컴포넌트 기반 소프트웨어가 아닌, 분산시스템에 적용하는 방법의 연구가 필요하다, 또한, Eclipse 등의 통합 개발환경과 연동하는 기술의 개발이 필요하다.

참고문헌

[1] OMG, " OMG Unified Modeling Language Specification, version 1.3, March 2000

[2] The Khronos Group Inc, " OpenMax Integration Layer API

Specification, version 1.1.2," Sept. 2008.

[3] Microsoft Corporation. DirectShow Online Documentation

[4] http://www.umlet.com/, UMLet, " OMG Unified Modeling Language Specification, version 1.3, March 2000