웹 애플리케이션을 위한 복잡도 척도

ᵁ이기열*, 이병정*, 이숙희**, 우치수*
* 서울대학교 컴퓨터공학부 **서경대학교 전자계산학과

Complexity Metrics for Web Application

ᵁKeeYoull Lee*, Byungjeong Lee*, Sukhee Lee**, Chi-Su Wu*
* School of Computer Science and Engineering, Seoul National University
** Dept. of Computer Science, Seokyung University

요 약

인터넷의 빠른 성장에 힘입어 웹 환경이 급속하게 성장하고 있다. 웹 환경은 TCP/IP 와 HTTP프로토콜을 이용한 분산 시스템 환경이고 클라이언트 서버 환경이다. 따라서 웹 환경에서 작동하는 웹 애플리케이션은 기존의 애플리케이션과는 다른 특징을 가진다.

현재 웹 애플리케이션의 개발이 많이 이루어지고 있다. 이에 웹 애플리케이션을 설계하고 개발하며 유지 보수하는 작업이 매우 중요해지고 있다. 그러나 최근의 웹 애플리케이션의 개발을 다양한 기술의 등장으로 인해 설계 및 유지 보수에 대한 충분한 고려 없이 빨리 개발되어 왔다. 이에 따라 웹 애플리케이션의 유지 보수가 어려워지게 되었다. 본 논문에서는 웹 애플리케이션을 정의하고 이의 복잡도 척도를 제안한다.

1. 서론

인터넷의 빠른 성장에 힘입어 웹환경이 급속하게 성장하고 있다. 웹 환경은 TCP/IP 와 HTTP프로토콜을 이용한 분산 시스템 환경이고 클라이언트 서버 환경이다. 따라서 웹 환경에서 작동하는 웹 애플리케이션은 기존의 애플리케이션과는 다른 특징을 가진다.

현재 웹 애플리케이션의 개발이 많이 이루어지고 있다. 이에 웹 애플리케이션을 설계하고 개발하며 유지 보수하는 작업이 매우 중요해지고 있다. 그러나 최근의 웹 애플리케이션의 개발을 다양한 기술의 등장으로 인해 설계 및 유지 보수에 대한 충분한 고려 없이 빨리 개발되어 왔다. 이에 따라 웹 애플리케이션의 유지 보수가 어려워지게 되었다. 본 논문에서는 웹 애플리케이션을 정의하고 이의 복잡도 척도를 제안한다.

2. 관련연구

2.1 웹 애플리케이션 복잡도 척도[1]

웹 애플리케이션에 대한 척도는 많이 제안되어 왔지만 현재까지는 웹 애플리케이션에 대한 척도이라기 보다는 협의의 하이퍼 미디어 프로세스와 프로덕트에 대한 척도가 제안되었다. 이 논문에서는 웹 애플리케이션의 범주를 HTML문서의 집합으로 정의하고 웹 애플리케이션을 저작하는 데에 필요한 개발 노력을 측정하는 척도를 제안하였다. 이 논문은 웹 애플리케이션의 유지 보수성, 웹 문서의 재사용성, 웹 애플리케이션 저작에 필요한 개발 노력을 측정하는 척도를 제안하고 경험적으로 검증한다.

Hyperdocument size, Connectivity, Compactness, Stratum, 이렇게 네 개의 척도가 제안되었는데 각각 하이퍼 문서의 크기, 링크의 수, 문서들이 내부적으로 상호 어떻게 연결되었는지, 웹 애플리케이션을 읽기까지 몇 단계를 거쳐야 하는지를 나타낸다. 기존의 개발 사례를 이용해 개발 노력과 주어진 척도와의 상관관계를

조사한 결과 Stratum을 제외한 나머지 세 척도가 양의 상관관계를 보였다.

2.2 웹 애플리케이션 설계 모델링[2]

웹 애플리케이션은 단순히 HTML문서의 집합이 아니라 사용자의 입력에 의해 서버의 비즈니스 로직에 영향을 미치는 소프트웨어로 볼 수 있다. 웹 애플리케이션은 크게 페이지, 폼, 컴포넌트, 프레임과 그 밖의 컴포넌트로 구성된다. 페이지는 웹 애플리케이션의 기본 부분이며. 페이지는 HTML형식의 정적인 페이지와 스크립트를 포함하는 동적인 페이지의 조합이다. 동적인 페이지에 포함되는 스크립트는 크게 서버 스크립팅과 클라이언트 스크립팅으로 나뉜다. 서버 스크립팅은 ASP,PHP,JSP등이 있고, 클라이언트 스크립팅은 Java Script, VB Script 등이 있다. 폼은 사용자로부터 입력을 수집하는 가장 일반적인 기술이다. 컴포넌트는 서버 컴포넌트와 클라이언트 컴포넌트를 생각할 수 있는데 서버 컴포넌트는 서버 상에서 동작하는 중간 층 컴포넌트이다. 이것은 대체로 컴파일된 프로그램으로 독립적인 비즈니스 로직을 수행한다. 클라이언트 컴포넌트는 자바 애플릿과 ActiveX 컨트롤이 있다. 프레임은 여러 페이지를 한 브라우저 안에 가질 수 있도록 한다.

3.1 웹 애플리케이션 아키텍처

본 논문에서는 웹 애플리케이션을 단순히 웹 문서의 집합이 아닌, 정적인 HTML페이지와 스크립트와 컴포넌트를 포함하는 동적 페이지의 집합으로 간주한다. 웹 애플리케이션을 이루는 페이지를 다음과 같이 분류한다.

(1) 정적 페이지 : 정적인 HTML 문서만을 포함하는 페이지이다. 이 페이지는 스크립트와 컴포넌트는 포함하지 않는다. 따라서 이 페이지는 다른 페이지에 비해 덜 복잡하고 따라서 개발 노력과 비용이 적게 드는 페이지이다.

(2) 클라이언트 스크립트 페이지 : 자바 스크립트와 비베스크립트와 같이 클라이언트 사이드에서 브라우저에 의해 해석되는 스크립트만을 포함하는 페이지이다. 이런 클라이언트 스크립트의 경우에는 복잡한 비즈니스 로직을 필요로 하지 않고 단순한 로직으로 클라이언트에서 실행되는 스크립트를 포함한다. 정적 페이지에 비해서는 더욱 많은 개발 노력과 비용을 필요로 한다.

(3) 서버 스크립트 페이지 : ASP나 PHP, JSP와 같이 서버에서 사용자의 입력에 반응하여 서버의 상태를 변화시키고 비즈니스 로직을 수행하는 페이지이다. 이 페이지는 데이터 베이스 서버와의 연동을 통해 트랜잭션을 수행할 수도 있다. 데이터 베이스에서 트랙잭션을 수행할 수도 있기 때문에 이 페이지는 클라이언트 스크립트보다 복잡하고 더 많은 개발 노력과 비용을 필요로 한다.

(4) 클라이언트 컴포넌트 페이지 : 자바 애플릿이나 ActiveX 컨트롤을 포함하는 페이지이다. 이 컴포넌트들은 클라이언트 사이드로 다운로드 되어 독립적으로 수행된다. 이 컴포넌트들도 데이터 베이스 서버와의 연동을 통해 트랜잭션을 수행할 수도 있다. 이 페이지는 컴포넌트가 컴파일러에 의해 컴파일된 객체이기 때문에 스크립트 페이지보다는 좀 더 복잡하고 더 많은 개발 노력과 비용을 필요로 하게 된다.

(5) 서버 컴포넌트 페이지 : 서버 컴포넌트를 포함하는 페이지이다. 이것은 서버에서 컴파일된 독립적인 프로세스가 수행되고 그 결과로 페이지가 보여지게 된다. 이 프로세스들은 각각 컴파일러 의해 컴파일된 객체이기 때문에 상당히 복잡하고 제일 많은 개발 노력과 비용을 필요로 하게 된다.

위의 다섯 개의 페이지를 기본으로 해서 웹 애플리케이션의 복잡도 척도를 측정한다. 그런데 클라이언트 스크립트와 서버 스크립트를 동시에 포함하는 경우가 있을 수 있는데 이런 경우에는 각각의 독립된 페이지를 갖는 두 개의 다른 페이지가 있는 것으로 간주하여 페이지를 척도에 반영한다. 예를 들어 다음 그림과 같은 경우를 보자.

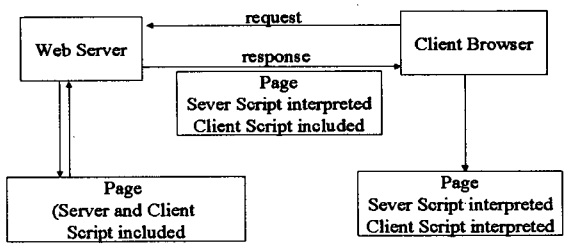

그림 1 서버와 클라이언트 스크립트가 포함된 페이지

클라이언트 스크립트와 서버 스크립트가 같이 있을 경

우 예를 들어 ASP의 경우에는 먼저 서버 스크립트가 서버에 의해 해석된 후에 그 결과 페이지가 클라이언트로 다운로드 되어 다시 웹 브라우저에 의해 클라이언트 스크립트가 해석되는 과정을 거친다. 따라서 각각의 페이지를 따로 고려하면 된다.

3. 2 복잡도 척도

위의 정의한 페이지들을 기본으로 해서 웹 애플리케이션을 위한 복잡도 척도를 정의하면 다음과 같다.

복잡도 척도 CM은

$$CM = k_1 * SP + k_2 * CSP + k_3 * SSP + k_4 * CCP + k_5 * SCP$$

SP : 정적 페이지의 수
CSP : 클라이언트 스크립트 페이지의 수
SSP : 서버 스크립트 페이지의 수
CCP : 클라이언트 컴포넌트 페이지의 수
SCP : 서버 컴포넌트 페이지의 수

여기서 k_1, k_2, k_3, k_4, k_5는 각 페이지의 특성에 따른 가중치 상수이다. 이 상수는 기존의 개발된 웹 애플리케이션을 통한 사례 연구를 통해 결정될 수 있다. 또한 웹 애플리케이션 개발자의 경험에 의해 개발 될 수도 있다.

두 번째 복잡도 척도는

$$CM_2 = \sum L_i$$

단 L_i 는 서버와 클라이언트에서 스크립트와 컴포넌트가 처리된 후의 i번째 결과 페이지의 링크의 수

이 척도는 웹 애플리케이션의 복잡도를 웹 애플리케이션이 가지고 있는 다른 페이지로의 총 링크의 수로 정의하였다. 웹 애플리케이션에서 링크가 많으면 많을수록 복잡도가 증가하게 되고 개발하는데 더 많은 비용과 시간을 필요로 한다.

4. 결론 및 향후 연구 과제

기존의 웹 애플리케이션 척도는 정적인 웹 문서만을 고려하여 현재의 웹 애플리케이션 기술의 변화를 반영하지 못하고 있다. 따라서 현재의 웹 애플리케이션 기술을 고려하여 웹 애플리케이션을 정의하고 이에 맞는 복잡도 척도를 새로 정의하였다. 새로운 복잡도 척도는 정적인 페이지뿐만 아니라 동적인 스크립트와 컴포넌트의 요소를 모두 포함하고 있다. 하지만 각 페이지마다 유저 인터페이스의 복잡성에 따라 다른 가중치를 가질 수 있는데 이 점은 고려 대상에서 제외하였다.

향후 새로운 척도를 경험적으로 검증하는 작업이 필요하다. 이것은 기존 웹 애플리케이션 개발 사례를 통해 척도의 타당성을 검증할 수 있을 것이다. 또한 웹 애플리케이션 설계 문서를 UML로 모델링하게 되면 UML문서로부터 제안된 척도 값을 자동으로 얻어 낼 수 있는 도구를 개발할 수 있다.

5. 참고 문헌

[1] E. Mendes, "Investigating Metrics for a Development Effort Prediction Model of Web Applications", *Proceedings 2000 Australian Software Engineering Conference*, pp.31-41.

[2] Jim Conallen, "Modeling Web Application Design with UML", Rational Software, http:// www.rational.com/uml/resources/whitepapers/ , 1998

[3] Jim Conallen, "Modeling Web Application Architecture with UML", *Communication of the ACM*, vol 42. no 10, Oct, 1999

[4] Eun Sook Cho, Soo Dong Kim, Sung Yul Rhew, "Object-Oriented Web Application Architectures and Development Strategies", *IEEE Internet Computing*, vol.3, no.1, Jan.-Feb. 1999, pp.60-8.