

## 동적 웹 어플리케이션의 특성을 반영한 조정 기능 점수 산정 방안

허승현<sup>○</sup>, 최은만  
 동국대학교 컴퓨터공학과  
 seknoj@hanmail.net<sup>○</sup>, emchoi@dgu.edu

## Adjusted Function Point Estimation Based on Characteristics of Dynamic Web Application

Seung Hyun Heo<sup>○</sup>, Eun Man Choi  
 Department of Computer Engineering, Dongguk University

### 요 약

소프트웨어의 측정은 소프트웨어의 생명주기 초기에 분석되어 규모와 비용의 예측에 도움을 주어야 한다. 초기에는 정확한 LOC 기반 규모예측이 어려워 기능 점수 기반의 측정에 의하여 예측할 수 있다. 그러나 현재의 기능 점수 기반 측정은 모든 시스템에 획일화 되어 있어 시스템의 특성을 반영할 수 없으며, IFPUG에서 제시한 일반 시스템 특성은 웹 기반 어플리케이션에서 사용자의 인지도가 매우 낮다. 본 논문에서는 웹 기반 어플리케이션 중에 동적 웹 사이트에 특화된 시스템 특성 집합 및 영향도 측정 방법을 제시한다. 이 요소를 근거로 동적 웹 사이트의 시스템 특성을 분석하여 기능 점수의 값 조정 인자에 반영하고 실제 시스템을 대상으로 조정된 기능 점수를 산정한다.

### 1. 서 론

소프트웨어의 비용 산정이나, 정확한 규모 예측은 소프트웨어 개발을 위한 팀 규모나 일정을 계획하고 전개시키는데 매우 중요하다. 이를 위한 연구가 30년 이상 진행되어 왔으며[1], 많은 방법들이 소개되고 사용 되어왔다. 대부분의 연구에서 소프트웨어의 정량적 측정을 위한 요소로 사용하는 것은 LOC(Line Of Code)이며, 현재까지 실무에서 가장 많이 사용되고 있는 요소이기도 하다. 그러나 LOC 기반의 측정은 어느 정도의 구현이 이루어진 후에야 정확한 측정을 할 수 있기 때문에 소프트웨어 생명주기의 설계 단계에서 규모나 비용을 예측하기에는 어려움이 있다. 또한 언어의 특성이나 프로그래머의 코딩 스타일에 종속적이기 때문에 많은 제한이 따른다.

기능 점수 매트릭은 사용자의 기능적 요구를 기반으로 측정하는 것으로서 초기에 더 정확한 예측을 할 수 있으며, 변화된 요구의 영향을 의뢰자에게 설명하기 쉽게 한다. 또한 소프트웨어 개발에 사용하는 어떤 언어나 틀에 대해서도 독립적이고, 비용, 노력, 결함 등의 데이터를 표준화시킴으로서 시스템과 시스템간의 비교평가를 가능하게 하고, 더 객관적인 평가를 하게 한다[2].

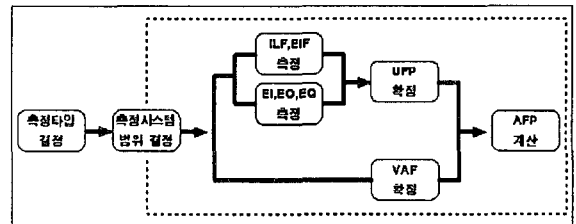
1990년대 중반 이후 넷스케이프의 보급과 함께 웹 기술은 급속하게 발전하였고, 높은 수준의 인터넷 환경이 각 가정까지 구축되면서 웹 비즈니스에는 다양한 요구와 복잡한 기술이 따르게 되었다. 하지만, 이들은 기존의 독립된 소프트웨어와는 다른 특성을 가졌고, 따라서 테스트 등의 소프트웨어 공학 분야에서도 확장되어 연구되고 있다[3]. 소프트웨어 측정 분야도 마찬가지이며, 기존의 기능 점수 모델로 웹 기반 어플리케이션의 특성을 감안하기에는 매우 부족하다.

본 논문에서는 기존의 기능 점수 산정 방법을 설명하고, 기존의 기능점수 산정법을 웹 어플리케이션의 측정에 사용하는 경우 문제점을 지적한다. 그리고 웹 어플리케이션 중에서도 동적 웹 사이트의 특성에 적합한 조정인자 집합을 제시하여 문제를 해결하고, 실제 동적 웹 사이트들을 두 집합의 시스템 특성을 적용하여 값 조정 인자를 구하고, 조정 기능 점수를 산정, 비교하여 이 논문에서 제시한 조정인자가 더 적합함을 보여준다. 마지막으로 본 연구를 요약, 미래 연구 방향을 제시한다.

### 2. 관련 연구

#### 2.1 기능 점수 산정 프로세스

기능 점수를 산정하는데 사용되는 전체 프로세스는 (그림1)과 같이 7단계로 요약된다[4].



(그림 1) 기능 점수 산정 프로세스[5]

- 단계 1 : 기능 점수 값의 유형을 결정한다.
- 단계 2 : 계산 범위와 어플리케이션 경계를 확인한다.
- 단계 3 : 모든 자료기능(내부 논리 파일과 외부 인터페이스 파일)과 복잡도를 확인한다.
- 단계 4 : 모든 트랜잭션 기능(외부 입력, 외부 출력, 외부 조회)과 복잡도를 확인한다.
- 단계 5 : 미조정 기능 점수(Unadjusted FP) 값을 결정한다.
- 단계 6 : 값 조정 인자인 14개의 일반 시스템 특성 (General System Characteristics: GSC)을 결정한다.
- 단계 7 : 조정 기능 점수(Adjusted FP) 값을 계산한다.

#### 2.1 웹 점수

IFPUG는 웹 기반 개발을 웹 기반 어플리케이션과 웹 인터페이스를 가진 어플리케이션, 동적 웹 사이트, 정적 웹 사이트 이렇게 네 가지로 분류하고, 그 중 정적 웹사이트의 기능적 크기 측정을 위한 '웹 점수(Web-Points)'를 발표하였다[6]. 웹 점수는 HTML

페이지의 수와 각 페이지의 복잡도를 근거하여 측정되며, 복잡도는 각 페이지에 있는 단어의 수와 링크의 수, 텍스트가 아닌 요소의 수에 따라 결정된다.

3. 동적 웹 사이트에 특성화된 시스템 특성

시스템의 특성을 고려하지 않고 기능 점수를 산정하는 것은 일반화되기 어려운 오류를 범할 수 있다. 그럼에도 불구하고 아직 많은 사람들이 미조정 기능 점수를 조정 가능 점수로 사용하고 있으며 일부에서는 일반 시스템 특성을 배제시키고 있다. 이는 사용자가 일반 시스템 특성이 클라이언트-서버 환경이나 웹 환경 보다는 메인 프레임 환경에서 보다 잘 연계 된다고 인식하기 때문이다[4]. 일반 시스템의 특성은 웹 환경 시스템의 특성과는 거리가 있어, 기능 점수 산정에 적용할 경우 시스템의 특성에 따른 기능 점수 산정이 어려울뿐더러 웹 시스템에 적용하기에 모호한 기준을 잘못 적용할 경우 오히려 악영향을 끼칠 수도 있다. 따라서 웹 환경에 특화된 시스템의 특성을 발견하고 그것들에 근거한 VAF를 산출하는 방법의 연구가 필요하다.

연구 방법에 있어 IFPUG에서 제시한 14가지 일반 시스템 특성에 대하여 다음과 같은 세 가지 규칙을 적용시켰다.

1. 웹 환경 시스템이라면 극단적인 점수가 부여될 수밖에 없는 요소는 제외시킨다.

: 이러한 특성은 다양한 웹 환경 시스템에 따라 정도의 차이를 부여할 수 있는 것이 아닌 웹 환경 시스템 자체의 근본적인 성질에 해당하는 것이다. 따라서 값 조정 인자를 구하는데 있어 각 웹 환경 시스템이 가지는 특성의 영향력을 극대화 시킨다.

2. 다양한 웹 환경 시스템의 특성이 고려될 수 있는 요소를 추가하도록 한다.

: 웹 환경 시스템의 특성을 구체화시켜 값 조정 인자에 반영함으로써 웹 환경 시스템의 특성이 잘 반영된 기능 점수의 산정이 가능하다.

3. 일반 시스템의 특성 요소 중 개선이 필요한 요소에 대해서는 개선한다.

: 웹 환경 시스템의 특성이 고려될 수 있는 요소이지만 점수 부여에 있어 개선이 필요한 경우는 개선하였다.

연구 결과 두 개의 요소가 삭제되었고, 두 개의 요소가 수정되었으며, 두 개의 요소가 추가되었다.

'분산 자료 처리' 특성은 분산 또는 실시간 어플리케이션에 대해서만 점수 부여가 가능하므로 웹 환경 시스템은 점수를 부여 받을 수 없다. 또한 웹 사이트의 경우 설치의 부담은 개발을 의뢰한 당사자가 아닌 호스팅 업체의 몫이고 웹사이트의 설치 및 웹 서버의 설정은 절차가 대부분 일치하므로 '설치의 용이' 특성은 '분산 자료 처리' 특성과 함께 삭제하였다.

'운영의 용이성' 특성은 사이트의 관리자에 의한 사이트의 관리, 변경 및 유지보수가 어느 정도의 범주 안에서 어느 정도의 개입으로 가능한지를 나타내는 요소로 정의하고, 점수 부여를 <표 1>와 같이 변경하였다.

<표 1> 운영의 용이성에 따른 영향도

점수	영향도(DI)
0	소스 코드의 수정으로 변경 가능하다.
1	GUI를 통한 접근으로 정적 텍스트의 변경이 가능하다.
2	GUI를 통한 접근으로 정적 이미지의 내용, 위치 및 링크 변경이 가능하다.
3	GUI를 통한 접근으로 정적 페이지의 구조 변경이 가능하다.
4	GUI를 통한 접근으로 동적 페이지의 폼 컴포넌트 변경 및 연산 이 가능하다.
5	GUI를 통한 접근으로 동적 페이지의 콘텐츠 생성, 추가 및 삭제 가 가능하다.

'복수 거점' 특성은 복수 위치에 있는 사용자 조직을 위해 개발되는 정도를 기술한다. 개선된 특성에서는 '사용자 고려'로 용어를 변경한다. 영향을 주는 요소는 (그림 2)와 같으며 점수 부여는 <표 2>를 따른다.

<표 2> 사용자 고려에 따른 영향도

점수	영향도(DI)
0	요소의 어느 것도 해당되지 않는다.
1~5	하나의 해당 항목 당 1~4번 요소는 1점을, 5번 요소는 2점을 부여한다. 4번 요소와 5번 요소는 중복 해당 될 수 없다. 해당 요소에 따른 부여 점수를 합산 한다.

1. 사용자 접근의 수를 고려한 최적화가 요구된다.
2. 사용자의 브라우저 종류 및 버전을 고려한 최적화가 요구된다.
3. 사용자 해상도를 고려한 최적화가 요구된다.
4. 사용자의 시스템 설정 분석이 요구된다.
5. 사용자 환경에 자동 설치 어플리케이션이 요구된다.

(그림 2) 사용자 고려 영향 요소

동적 웹 사이트 중에서도 개발에 사용된 언어의 종류와 수, 서버의 종류와 수에 따라 시스템의 복잡도가 달라진다. 또한 시스템의 동적인 정도, 서비스의 이용자 특성화 정도에 따라 시스템 특성이 구분될 수 있다.

삭제된 일반 시스템 특성(GSC) 대신 다음과 같은 두 가지 특성을 추가 하였다.

'시스템 구성 요소' 특성은 시스템을 이루고 있는 환경과 시스템 개발을 위해 사용하는 언어에 관련된 것이며, 영향도에 따른 점수 부여는 <표 3>과 같다. <표 3>에서 개발 언어란 마크업 언어, 스크립트 언어, 어플리케이션 개발 언어 등 모두를 고려하여 정하였으며, 서버 구성 또한 동적 웹 사이트가 추가 될 수 있는 모든 환경을 고려하여 정하였다. 단 스크립트 언어는 종류에 상관없이 최대 1개 언어로 간주한다.

<표 3> 시스템 구성 요소에 따른 영향도

점수	영향도(DI)
0	로컬 서비스만 지원 가능하다.
1	한 개의 웹 서버로 구성되며, 한 개 이상의 언어로 개발되었다.
2	한 개의 웹 서버와 한 개의 데이터베이스 서버로 구성되며, 세 개 이상의 언어로 개발되었다.
3	한 개의 웹 서버와 한 개의 데이터베이스 서버로 구성되며, 네 개 이상의 언어로 개발되었다.
4	한 개의 웹 서버와 한 개의 데이터베이스 이외의 추가적인 서버가 존재하며, 세 개 이상의 언어로 개발되었다.
5	한 개의 웹 서버와 한 개의 데이터베이스 이외의 추가적인 서버가 존재하며, 네 개 이상의 언어로 개발되었다.

'서비스 개인화' 특성은 제공하는 서비스의 특성이 이용자 개인에 대하여 어느 정도 특성화 되어 있는지를 나타낸다. 동적 웹 사이트는 컴퓨터 분야의 응용 중에서도 하나의 응용에 특히나 사용자가 많은 분야이다. 또한 사용자에 따라 다른 동적인 환경을 제공한다. 이러한 특성은 시스템의 복잡도에 많은 영향을 준다. '서비스 개인화' 특성에 따른 영향도는 <표 4>와 같다. 예를 들면 점수 3은 회원 가입절차가 있고 회원 정보에 따른 동적인 페이지가 작성되는 경우이고, 점수 5는 회원 등급이 구분되고 회원 등급

에 따라 다른 환경과 서비스가 제공되는 경우라 할 수 있다.

<표 4> 서비스 개인화에 따른 영향도

점수	영향도(DI)
0	관리자와 사용자의 구분이 불가능하다.
1	관리자와 사용자의 구분이 가능하나, 사용자 개개의 구분이 불가능하다.
2	관리자와 사용자 개개의 구분이 가능하지만 사용자의 구분으로 동적인 페이지가 작성 되지 않는다.
3	관리자와 사용자 개개의 구분이 가능하고, 사용자에 따라 동적인 페이지가 작성된다.
4	관리자와 사용자 개개의 구분이 가능하고, 역할 또는 그룹이 구분되어 있지만 모든 역할과 그룹에 획일적인 서비스를 제공한다.
5	관리자와 사용자 개개의 구분이 가능하고, 역할 또는 그룹이 구분되어 있으며, 역할 또는 그룹에 따라 다른 서비스를 제공한다.

4. 사례연구

본 연구에서 제시한 기능 점수 산정 방법과 기존의 기능 점수 산정 방법을 비교하기 위해 실험 자료로 세 개의 시스템을 선택하였다. 첫 번째 실험 자료는 시스템 크기에 비해 비교적 처리능리 수준이 높은 웹 경매 시스템이다. 이 시스템은 윈도우 환경에서 개발되어, 윈도우 환경에서 수행되는 웹 경매 시스템이다. 6개의 테이블로 이루어진 데이터베이스와 관계하고, ASP 언어로 개발된 28개의 파일이 동적 처리와 관계한다.

다른 하나는 처리능력에 비해 데이터베이스 크기가 큰 웹 중고 서적 매매 시스템이다. 7개의 데이터 테이블과 72개의 ASP 파일, 4개의 HTML파일로 구성되고, 스타일 시트 파일 1개와 방문자 수 기록을 위한 하나의 text 파일을 포함하는 시스템이다.

세 번째 실험 자료는 기차표를 조회, 예약할 수 있는 온라인 서비스를 제공한다. 여러 곳에서 접속자가 동시에 표를 예약하려 할 수 있으므로 요청에 민감하다. JAVA와 서블릿으로 구현되었으며, 총 28개의 클러스터파일과 초기화면이 될 수 있는 6개의 HTML 파일로 이루어져 있다.

세 가지 시스템의 기능점수를 두 그룹의 조정인자를 적용해 산정하고, 이를 실제 개발단계에서의 소요시간과 비교해 보았다. 비교 방법은 시스템의 개발단계에서 소요된 M/M(Men/Month)과 미조정 기능점수, 일반 시스템 특성을 적용시킨 조정 기능점수, 웹 시스템 특성을 적용시킨 조정 기능점수와와의 상관계수(correlation)를 각각 구하였다.

실험에서 측정된 각 시스템의 기능 점수와 실제 개발에 소요된 M/M을 요약하면 <표 5>과 같다.

<표 5> 각 시스템의 기능점수와 M/M 산정 결과

	웹 경매 시스템	중고서적 매매 시스템	기차표 예매 시스템
미조정 FP	90	139	88
일반 시스템 특성을 이용한 FP	86.4	125.1	85.36
웹 시스템 특성을 이용한 FP	87.3	133.44	95.04
M/M(Men/Month)	1.6	1.93	1.8

<표 6 > M/M와 각 FP와의 상관계수

	미조정 FP	일반 시스템 특성을 이용한 FP	웹 시스템 특성을 이용한 FP
M/M	0.777518	0.784801	0.883182

이를 바탕으로 실제 개발에 소요된 M/M와 각각의 FP의 상관계수를 구해본 결과 <표 6>와 같다. 참고로 상관계수의 범위는 -1에서 1 사이이며 상관계수의 절댓치가 1에 가까울수록 상관이 깊고, 0에 가까울수록 상관이 적다는 것을 의미한다.

<표 6>에 보는 바와 같이 일반 시스템 특성을 이용한 조정 FP는 M/M와의 상관계수에 있어 미미한 상승효과를 볼 수 있을 뿐이다. 반면에 웹 시스템 특성을 이용한 조정 FP는 실제 개발에 소요된 M/M와의 상관계수가 매우 컸음이 확인된다. 또한 일반 시스템 특성을 이용한 값 조정 인자를 보면 성격이 다른 시스템임에도 불구하고 그다지 차이가 나지 않는 모습을 보여준다. <표 6>의 실험 결과에서도 M/M와의 상관계수가 많이 개선되지 않는 모습을 볼 수 있다. 반면 웹 시스템의 특성을 이용한 값 조정 인자의 산정 결과를 보면 조정인자의 의도대로 시스템의 특성이 많이 반영되었음이 확인된다.

5. 결론 및 향후 연구 과제

기능 점수를 기반으로 하는 측정 모델은 이미 세계적으로 인정되었고, 많은 적용이 이루어지고 있는 방법이다[4]. 기능 점수 산정 시스템의 특성을 반영하여 더 정확한 기능 점수를 산출 하고자 국제 기능 점수 사용자 그룹(IFPUG)에서는 일반 시스템 특성 14가지를 제시하여 값 조정 인자를 반영하게 하였다. 그러나 웹 기반 어플리케이션에 있어서 일반 시스템 특성은 기능 점수 사용자들에게 인식이 좋지 않다. 일반 시스템 특성은 웹 기반 어플리케이션 보다는 메인 프레임 어플리케이션에 적용이 잘 되기 때문이다[4]. IFPUG에서 웹 기반 개발을 네 가지로 분류하고 그 중 정적 웹 사이트에 대해서만 '웹 점수'라는 특화된 모델을 발표하였다.

본 연구에서는 웹 기반 어플리케이션, 그 중에서도 동적 웹 사이트를 대상으로 기능 점수의 시스템 특성 반영을 높이고자 하였다. 따라서 시스템의 특성을 반영하는 IFPUG가 제안한 일반 시스템 특성 14가지를 분석하였으며, 결과로 두 가지 일반 시스템 특성을 삭제하고, 두 가지 특성을 개선, 두 가지 특성을 추가하였다.

논문에서는 연구 결과로 나온 동적 웹 사이트를 대상으로 하는 시스템 특성 집합을 설명하였다. 그리고 4장에서는 기존의 시스템 특성과 새로 제안된 시스템 특성을 요소로 두 가지 기능 점수를 산정하였다.

기능 점수는 데이터베이스의 자료 기능과 요구 분석의 트랜잭션 기능에 의해 분석되고 계산된다. 따라서 기능 점수를 이용한 측정은 분석 또는 설계 단계에 가능하며, 이로써 충분한 장점을 갖추었다고 할 수 있겠다. 하지만 지금의 기능 점수 산정 방법은 여러 종류의 시스템에 특화되어 있지 않으며, 전혀 성격이 다른 시스템에도 비슷한 규칙을 적용할 수밖에 없는 실정이다.

앞으로, 웹 사이트 이외의 웹 기반 어플리케이션에 대해서도 특화된 기능 점수 산정 모델을 연구해야 하며, 웹 기반 어플리케이션이 아닌 다른 여러 특성의 시스템에 대해서도 특화된 산정 모델이 필요할 것이다. 더 나아가 많은 사용자에게 의한 적용으로 입증된 모델의 표준화 작업 역시 미래의 연구과제이다.

6. 참고문헌

- [1] J. E. Matson, B. E. Barrett and J. M. Mellichamp, "Software Development Cost Estimation Using Function Points", IEEE Trans. on Software Eng., Vol.20, No.4
- [2] Furey, S., "Why We Should Use Function Points", Software, IEEE, Vol.14, Issue: 2, Mar/Apr 1997 Pages:28, 30
- [3] 최은만, 웹 기반 소프트웨어의 시험 및 검증기술, 정보과학회지 Vol. 4, No. 3, pp. 139-150 (November 2001).
- [4] Garmus David, Herron David, "Function Points Analysis", Addison-Wesley, 2001
- [5] 한국 기능 점수 사용자 그룹, "기능 점수 측정 절차", www.kfpug.co.kr, 2004
- [6] IFPUG, "Web-Based Development and Functional Size Measurements", IFPUG 2000 conference, 2000