

# 소프트웨어 규모예측을 위한 기능점수모형의 개선

안 연 식<sup>†</sup>

## 요 약

본 논문에서는 기존의 기능점수모형을 보완하여 활용도와 정확도를 향상시킨 새로운 기능점수모형을 제안하였다. 제안된 모형에서는 기능유형을 7개로 정의하고, 기술적 복잡도는 일반적인 소프트웨어의 기술성과 분산 환경에 따른 구현의 난이도를 평가하는 간편한 방법으로 활용도를 높였다. 또한 이 모형의 정확도를 검증하기 위해 24건의 소프트웨어개발 프로젝트에 제안모형을 적용하여 산정한 총기능점수와 실무인력과의 회귀식을 도출하였다. 그 결과 제안모형이 기존의 기능점수모형보다 높은 수준의 기능점수모형임이 밝혀졌다.

## Improved Function Point Model for Software Size Estimation

Yeon-Shick Ahn<sup>†</sup>

### ABSTRACT

In this paper an improvement is made on the function point model, one of software size estimation model, for better feasibility with increased accuracy as my proposed model. For better feasibility seven function types are proposed in this model and technical complexity factors are used to assess the difficulties of considering technical and distributed environment attributes. To assess the creditability of the proposed model, 24 software development cases were considered in the regression model with the actual required efforts as the total function points. As seen in the statistical results, the proposed model is at least as superior to the known function point models.

### 1. 서 론

본 논문에서는 소프트웨어규모 예측모형중의 하나인 기능점수모형에 의한 소프트웨어규모 예측 방법과 이 모형에 대한 그 동안의 여러 가지 연구들을 고찰한 후, 소프트웨어개발 프로젝트에 쉽게 적용할 수 있도록 활용성을 높이면서도 규모 예측의 정확도를 향상시킨 개선된 기능점수모형을 제안하며 실제 자료를 수집하여 제안모형을 평가하도록 한다.

### 2. 기능점수모형

#### 2.1 기능점수모형의 개요

기능점수모형은 1980년대 미국 IBM의 A. J. Albrecht가 제시한 소프트웨어규모 예측모형으로, 소프트웨어를 구성하는 기능수에 가중치를 고려하는 방식이다.[1][2] 이는 소프트웨어규모를 원시코드수(Line of Source Code)를 이용하여 예측하는 COCOMO모형[3]과는 구별된다. 즉 소프트웨어가 근본적으로 자료의 입력, 정보의 가공, 정보의 저장 및 정보의 출력 시스템이라는 점을 중시하여 외부입력, 외부출력, 데이터화일, 인터페이스 등 기능유형의 양이 많으면 많을수록 규모가 커지므로 기능유형별 기능수를 구한

<sup>†</sup>정 회 원:경원전문대학

논문접수:1996년 5월 9일, 심사완료:1997년 4월 15일

뒤 여기에 해당 가중치를 곱하여 기능점수를 계산하고, 이를 소프트웨어규모의 예측 및 소요인력의 추정 에 이용한다.

2.2 기능점수모형의 특징

기능점수를 소프트웨어규모 예측에 활용하는 가장 중요한 이유는 기능점수를 계수하기 위해 개발 초기 단계에서 사용자와 개발자가 쉽게 의논할 수 있다는 점이다. 또한 기능점수모형은 사용자의 외부적 시각 으로부터 접근하여, 응용프로그램에서 제공하거나 사용하게 될 입력이나 출력을 파악하는 등 기술적 요소와는 독립적으로 비교적 쉽게 그리고 정확한 규모 예측을 가능하게 한다. 그리고 기능점수를 소프트웨어개발 생산성 즉 소요공수(인·월)당 기능점수 또는 기능점수당 소요공수 등을 산정하는 척도로 활용할 수도 있다.

반면에 적용 범위가 사무처리용으로 제한적이라는 점이 단점으로 지적되고 있으나, 국내의 소프트웨어 개발 프로젝트가 대부분 사무처리용 임을 감안하면 본 모형은 매우 유용한 것이라고 할 수 있다.

<표 1>COCOMO모형과 기능점수모형의 비교  
<Table 1> Comparison between COCOMO and function point model

구분	COCOMO	기능점수모형
창시자	B. W. Boehm	A. J. Albrecht
가치기준	원시코드 수	기능 수
생산성 측정요소	원시코드 수 (스텝 수)	기능유형별 기능수
소요공수 산정인자	보정계수 (15종)	기술적 복잡도 (14종)
적용범위	다목적	사무처리용

2.3 기능점수모형에 의한 소프트웨어 규모 예측

전형적인 기능점수모형에 의하여 소프트웨어 규모를 예측하는 방법은 다음과 같다.[1]

첫째, 소프트웨어의 기능을 증대시키는 각 기능유형들에 대하여 기능수를 측정하고 기능의 난이도에 따른 적절한 가중치를 선택하여 곱함으로써 기본기능점수를 구한다.

둘째, 프로젝트에서 요구하는 성능 및 품질 특성에

대한 기술적 복잡도를 계산하고, 기본기능점수에 이 기술적 복잡도에 따른 영향도를 곱하여 총기능점수를 구한다.

셋째, 총기능점수를 이용하여 원시코드수 또는 소요공수를 구하고 개발비용을 산정한다. 또한 총기능점수를 기타 생산성의 척도로 활용할 수도 있다.

2.3.1 기능유형별 기능수 측정

Albrecht의 연구[2]에서 정의된 기능유형은 5종으로서, 개발 대상 소프트웨어가 명확히 정의되고 외부 시스템과의 인터페이스가 분명해진 상태에서 기능유형별 기능수를 다음과 같이 측정한다.

① 외부입력(External Input Types)

소프트웨어로 구성되는 시스템의 경계 밖에서 시스템에 데이터를 입력하는 기능을 말하며, 내부논리화일에 내용이 추가되거나 내용 변경을 수반하는 입력 기능을 의미한다. 사용자가 직접 데이터를 입력하는 경우만이 아니라 다른 시스템이 작성한 자료를 자기테이프(Magnetic Tape)나 통신회선을 통하여 입력하는 경우도 포함한다. 또한 형식이 다르거나 처리절차가 상이한 입력도 별도의 기능으로 취급한다. 그러나 외부입력 기능이지만 외부조회 또는 외부연계용 화일에서 계수되는 것과, 사용되는 기술(technology) 때문에 포함되는 기능은 제외한다.

② 외부출력(External Output Types)

시스템의 경계밖으로 즉 대개 사용자나 다른 프로그램 등으로 데이터를 출력하는 기능의 수를 말한다. 화면이나 프린터 출력은 물론 자기테이프와 같은 2차 출력을 포함한다. 또한 형식이 다르거나 처리절차가 상이한 출력도 별도의 기능으로 취급한다. 또한 외부 입력에서와 마찬가지로 외부출력 기능이지만 외부조회나 외부연계용화일에서 계수되는 것과 사용되는 기술때문에 포함되는 기능은 제외한다.

③ 내부논리화일(Logical Internal File Types)

시스템내에 논리적으로 유지되는 화일로서 사용자 관점에서 볼 때 시스템에 의해 생성되고 사용되며, 또한 관리되는 논리적인 데이터의 집합체이다. 그러나 사용자가 외부입력, 외부출력, 연계용화일 및 외부

조회를 통하여 접근할 수 없는 내부논리화일은 제외한다.

④ 외부연계용화일(External Interface File Types)

응용시스템간에 공유되거나 전송되는 화일로써 시스템밖의 응용시스템이 논리적으로 유지하는 화일의 수를 말한다.

⑤ 외부조회(External Inquiry Types)

외부조회는 어떤 입력이 즉시 출력을 요구하는 일

〈표 2〉 기능유형별 복잡도의 평가 기준  
 (Table 2) Complexity evaluation criteria

기능유형	가중치 평가 기준		가중치
외부입력	단순	자료 항목의 수가 매우 적은 경우, 사용자 인터페이스에 대한 고려가 설계에 많이 영향을 미치지 않음	3
	보통	단순하지도 복잡하지도 않음	4
	복잡	많은 자료 항목이 있는 경우, 사용자 인터페이스에 대한 고려가 설계에 영향을 많이 미치게 됨	6
외부출력	단순	1-2열(column)정도의 짧은 출력, 단순한 데이터, 변형(transformation)의 경우	4
	보통	소계가 있는 복수개 열의 출력, 단순한 데이터 변형의 경우	5
	복잡	복잡한 자료 항목과 변형의 경우	7
내부 논리 화일	단순	레코드와 자료 항목의 수가 매우 적은 경우, 성능이나 복구에 대한 고려가 심각하지 않음	7
	보통	단순하지도 복잡하지도 않은 경우	10
	복잡	레코드와 자료 항목이 많은 경우, 성능이나 복구에 대한 고려가 심각함	15
외부 연계용 화일	단순	레코드와 자료 항목의 수가 매우 적은 경우, 성능이나 복구에 대한 고려가 심각하지 않음	5
	보통	단순하지도 복잡하지도 않은 경우	7
	복잡	레코드와 자료 항목이 많은 경우, 성능이나 복구에 대한 고려가 심각함	10
외부조회	단순	외부입력 및 외부출력기준을 적용하되 둘 중에서 큰 값을 적용	3
	보통	외부입력 및 외부출력기준을 적용하되 둘 중에서 큰 값을 적용	4
	복잡	외부입력 및 외부출력기준을 적용하되 둘 중에서 큰 값을 적용	6

력·출력 조합을 말한다. 입력 또는 출력 중의 하나라도 형식(format)이 다르거나 내부 처리 절차가 다르면 별개의 기능으로 간주한다. 외부조회를 위한 입력은 내부논리화일에 있는 데이터의 탐색만을 일으키는 점에서 데이터 갱신을 수반하는 외부입력과는 구별된다.

2.3.2 기본기능점수 계산

기본기능점수는 기능유형별 기능수에 〈표 2〉의 기능유형별 3단계 가중치중에서 적합한 가중치를 선정하여 곱한 값으로, 추후 기술적 복잡도에 의한 보정을 거쳐 최종적인 기능점수로 환산된다.

기본기능점수(FPu)의 총계는 (식 1)과 같이 계산된다.

$$FPu = \sum_{i=1}^5 (FC_i * W_i) \quad (1)$$

여기서 FPu = Unadjusted Function Point(기본기능점수), FC<sub>i</sub> = Function Count(기능유형별 기능수), W<sub>i</sub> = Weight(기능유형별 3단계 가중치)이며 가중치의 평가 기준은 〈표 2〉와 같다.

2.3.3 총기능점수 계산

총기능점수는 기본기능점수(FPu)에 기술적 복잡도(Technical Complexity Factor)를 고려하여 시스템의 최종적인 기능점수로 산출되는 결과치이다. 기술적 복잡도 평가항목과 그 영향도를 판정하는 기준은 〈표 3〉과 같다.

기술적 복잡도의 14개 평가항목당 영향도를 0에서 5까지 가중치를 주고 평가하여 합산하면 그 결과는 0부터 70의 범위의 값이 되므로 (식 2)에 의해 기술적 복잡도에 따른 영향도(Total Degree of Influence)를 이용하여 ±35% 범위 내에서 보정된 총기능점수(FP)가 계산된다.

$$FP = FPu * (0.65 + 0.01 * TDI) \quad (2)$$

2.3.4 소프트웨어 규모의 예측

총기능점수를 이용하여 소프트웨어의 규모를 예측하는 방법은 기능점수당 원시코드수를 환산하는 공식이나 또는 기능점수와 소요공수와의 회귀식을 이

〈표 3〉 기술적 복잡도 평가항목  
 (Table 3) Technical complexity factors

평가항목	평가내용
Data통신	데이터 통신의 필요 정도
기능분산	분산 처리 기능의 정도
시스템 성능	응답 속도나 처리율(throughput)과 같은 시스템의 성능(performance) 요구
운영환경	운용될 시스템의 여유 정도(사용 정도)
온라인처리	온라인 트랜잭션 빈도
온라인 Data입력	온라인 자료 입력 및 제어 기능 요구 정도
Interface 복잡성	온라인 입력이 복수 개의 화면 또는 오퍼레이션을 사용하는 트랜잭션으로 구성되는 정도
온라인 Data갱신	내부논리화일을 온라인 갱신하는 정도
처리의 복잡성	많은 제어 상호작용과 의사 결정의 수반, 광범위한 논리, 수학적 사용 및 많은 예외 처리로 인한 불완전한 처리 등 처리의 복잡성 정도
제사용성	프로그램 및 내부 코드가 다른 프로그램에서 혹은 다른 지역에서 제사용의 고려 정도
설치용이성	변환 및 설치 용이성을 고려하는 정도
신뢰성	효과적인 기동, 백업, 복구 등 운용성의 관리함을 위해 프로그램이 수작업(테이프, 서식 작성)을 최소화한 고려 정도
다중설치성	복수의 조직을 위한 복수 장소에서의 설치를 고려하는 정도
유지보수 용이성	변경의 용이성을 위해 유연한 질의어 기능이나 사용자에 의한 테이블 수정 기능 등을 고려하는 정도

용한다.

### 3. 관련 연구 고찰

기능점수모형은 사용자 관점에서 소프트웨어 규모를 예측하는 유용한 방법이지만 기존의 모형들은 적용 범위, 측정의 신뢰성, 가중치의 적용 등에서 여러 가지 문제점도 내포하고 있다. 따라서 기능점수모형에 관련된 연구는 모형 자체의 개선, 활용성의 향상, 그리고 적용 범위의 확대 등 3가지 방향으로 진행되고 있는데, 주요 연구결과를 요약하면 다음과 같다.

#### 3.1 IFPUG(International Function Point User Group)

##### 모형

IFPUG는 미국에 본부를 둔 기능점수모형 관련 국

제사용자단체로서 지속적으로 기능점수모형을 보완하여 활용성을 높이는 연구를 수행하고 있다.

IFPUG모형은 Albrecht의 모형에서 기능수를 측정하는 방법을 보다 심도있게 개선한 방법이다. 즉, 기능점수모형의 근간이 되는 5개 기능유형은 그대로 유지하면서 각 기능의 복잡도를 평가하는 기준을 체계화시켜 정확성을 향상시키고 있는데, 기능의 복잡도는 기능이 다루고 있는 데이터요소의 수와 입출력 기능에서 참조하는 화일수, 또는 화일이 포함하는 레코드의 수에 의해 결정된다. 외부입력, 외부출력, 외부조회는 화일수를 사용하여 복잡도를 결정하고 내부는리화일 및 외부연계용화일은 레코드의 종류를 사용한다.

#### 3.2 Symons의 Mark II 모형

Symons[4]은 Albrecht의 기능점수모형에서 의 기능의 복잡도를 단순, 보통, 복잡 등의 3단계로 구분하는 것, 기능의 가중치 설정부분, 기능유형에서 내부 처리의 복잡도 제외, 기능점수 합산시의 불합리, 그리고 기술적 복잡도 평가상의 문제점을 지적하고 이를 개선하는 방법으로 (식 3)과 같이 Mark II 방법이라는 기능점수 계산 공식을 제안하였다.

$$FC = 0.44 N_I + 1.67 N_E + 0.38 N_O \quad (3)$$

여기서 FC=조정하기 전의 기능점수

$N_I$  = 입력 데이터 유형의 수

$N_E$  = 참조되는 Entity Type의 수

$N_O$  = 출력 데이터의 수

이며, 기능수를 조정하기 위하여 기존의 14개 복잡도 평가항목에 아래와 같은 6개의 항목을 추가하여 20개의 요소를 사용하고 있다.

- 다른 응용시스템과의 인터페이스 정도
- 특별하게 요구되는 보안 기능 정도
- 제 3자의 직접 접근 요구 정도
- 문서화에 대한 요구 정도
- 교육 훈련 하부 시스템 등과 같이 특별하게 요구되는 사용자 교육 훈련 요구 정도
- 사용자가 특별히 요구하는 하드웨어나 소프트웨어를 정의, 선택, 설치하는 요구 정도

3.3 기능점수 측정의 신뢰성 향상을 위한 연구

Kemerer & Porter[5]는 기능점수 측정상의 주요 오차요인 즉 백업화일을 세는 방식, 에러메시지를 세는 방식 등 14가지의 질문을 통하여 3개 조직에서 기능점수의 변화를 분석한 바 비교적 소수의 요소만이 기능점수 측정치의 오류수준에 영향을 미친다고 분석하였다.

또한 Low & Jeffery[6]는 기능점수 측정에 대한 실험을 수행하여 다음과 같은 결과를 도출하였다.

- 한 조직 내에서 기능점수를 사용할 때, 측정치의 오차는 30% 이내인 것으로 분석된다.
- 서로 다른 조직에서 기능점수를 사용할 경우 조직간에 측정값의 오차가 생길 수 있다.
- 기능점수를 잘 적용하기 위해서는 적용 경험이 필수적이다.
- 소프트웨어개발 경험이 많은 사람의 측정치는 기능점수 활용 경험이 많은 사람과 비슷한 정확성을 보인다.

3.4 기능점수모형의 적용 범위 확장 연구

원래 기능점수모형이 입출력을 중심으로 하는 사무처리시스템에 적용하는 목적으로 개발되었기 때문에 적용 범위에 대한 확장이 필요하다.

Jones[7]는 기능점수모형을 변형하여 특성 점수(Feature Point)를 개발하고, 내부 처리 알고리즘이 복잡한 소프트웨어, 즉 실시간용 소프트웨어, 프로세스제어용 소프트웨어, 장비내장형 소프트웨어 등의 규모를 견적하는데 유용하게 활용될 수 있는 특성점수모형을 제안하였다.

Mukhopadhyay & Kekre[8]는 기능점수모형을 프로세스중심 시스템에 적용하기 위해 프로세스 제어 기능을 프로세스의 커뮤니케이션특성(CF; Communication Feature), 위치를 제어하는 위치특성(PF; Position Feature), 그리고 동작을 제어하는 동작특성(MF; Motion Feature)으로 구분하여 (식 4)에 의해 기능수(Function Count)를 산정하는 모형을 제안하였다.

$$\text{기능수} = 25.61 + 2.37(\text{MF}) + 9.66(\text{TF}) + 28.39(\text{CF}) + 5.75(\text{MF} * \text{PF}) \quad (4)$$

3.5 일본의 기능점수모형

일본에서는 다음과 같이 원래의 기능점수모형을 보다 간편하게 개선한 간이기능점수법을 주로 사용하고 있으며, 대기업을 중심으로 IFPUG의 기능점수모형을 개선시킨 개량기능점수법 동도 널리 활용하고 있다.[9][10]

3.5.1 COSDES 모형

일본에서는 1985년 COSDES(Committee on Software Development Estimation System)가 설립되어 소프트웨어 규모 예측에 대한 연구의 일환으로 기능점수모형을 연구하고 있다. 현재 COSDES의 기능점수모형은 Albrecht의 모형을 단순화하여 기능점수를 계산하고, 계산된 기능점수에 의해 소프트웨어 규모를 원시코드수로 예측하여 개발비를 산정하는 구조로 되어 있다.

기능유형으로는 화면, 출력장표, 화일로서 화면은 다시 입력, 출력, 입출력, 메뉴로 세분되고, 화일은 입력, 출력, 입출력으로 세분하여 기능점수를 계산한다. 각각의 가중치와 소프트웨어규모 예측은 (식 5)에 의한다.

$$\text{KS(천step)} = 6 + 0.09 * \text{FP} \quad (5)$$

3.5.2 간이기능점수법

간이기능점수법은 예를 들면 Client/Server환경에서와 같이 사용하는 기술과 시스템의 특성으로 보아 원시코드수 방식이 적합하지 않은 경우를 위해 개발되었다. 기능유형으로는 화면과 장표만으로 한정하여 간편하게 개발규모를 추정하는 모형이다.

3.5.3 개량기능점수법

개량기능점수법은 간이기능점수법보다 정밀도가 높고, 본래의 기능점수법보다 사용하기 간편한 방법으로 일본IBM 등에서 사용하는 자료항목수에 의한 규모예측법이 대표적인 사례이다. 일본IBM에서는 SI사업의 견적을 위해 자료사전에 나타나는 자료항목수를 기준으로 소프트웨어의 규모를 예측하며, 약 3,000건에 이르는 개발 사례로부터 축적된 데이터를 바탕으로 모형을 구축하여 활용하고 있다.

4. 개선된 기능점수모형의 제안 및 평가

4.1 제안 배경 및 기존 모형의 문제점

기능점수모형은 기본적으로 개발된 소프트웨어를 사용하게 되는 최종사용자의 필요에서 출발한다고 볼 수 있으며, 이러한 사용자의 필요성을 충족시킨다는 관점에서 소프트웨어의 핵심은 그 기능이다. 따라서 기능점수모형은 소프트웨어개발 목적과 부합하는 합리적인 방법으로 인식되고 있다. 그러나 국내에서 기능점수모형이 현재 널리 사용되지 못하고 있는 이유는 다음과 같다.[11]

첫째, 기능점수모형에 대한 인식의 부족이다. 일반적으로는 프로그램의 본수나 원시코드수를 기준으로 산정하는 방식에 익숙하며 기능유형의 종류와 기능유형의 양 그리고 난이도 판정을 필요로 하는 기능점수모형을 사용하는데 익숙하지 않다.

둘째, 기술적 복잡도 계산에 있어 영향도를 주관적으로 판정할 수밖에 없어 명확한 기준을 요하는 경우를 만족하지 못하고 있다.

이밖에도 기능점수모형의 활용도 및 정확도를 제고하기 위하여 개선해야 할 점은 다음과 같다.[12] 첫째, 모든 시스템 구성요소인 기능유형의 복잡도 구분을 '단순, 보통, 복잡'으로 한정된 것이 지나치게 단순하여 정확도가 저하될 수 있다. 둘째, 가중치가 특정 조직에서가 아닌 다양한 개발 환경에 적용될 수 있는지 검증이 필요하다. 셋째, 기능점수는 Albrecht가 주장한 대로 단순히 합산하여 구하면 정확하지 못한 결과를 내기도 한다. 넷째, 14개의 기술적 복잡도 평가항목 중에 중복되는 요인이 있으며 영향을 미치는 정도의 가중치가 너무 단순하다.

4.2 개선된 기능점수모형의 제안

본 논문에서는 이와 같은 문제점을 고려하고 기능점수모형의 정확도와 활용성을 높이기 위해 다음과 같은 기능점수모형을 제안한다.

첫째, 측정 대상 기능유형의 종류를 가능한 적게 한다. 즉 기능유형을 입력, 처리, 그리고 출력중심으로 하되 한 화면에서 입출력 기능을 동시에 수행하는 온라인 처리의 일반적인 특성을 고려하여 입출력 부분을 공유할 수 있도록 기능유형을 정의하였으며, 처리 부분은 개발 초기 단계에서 기능수의 측정이 어려운 점을 고려하여 처리 기능에서 참조되는 데이터베이스 또는 화일수를 측정하여 가중치로서 조정하도록 한다.

둘째, 기존의 모형에서 세분화되어 있는 기술적 복잡도 평가항목을 분류체계에 의해 집산화하여 활용성을 높인다.

셋째, 최근 들어 국내의 소프트웨어개발 환경의 추세인 Client/Server환경에서의 소프트웨어규모 예측에도 적용 가능한 기능점수모형을 제안한다.

4.2.1 기능유형에 관한 개선

(1) 측정 대상 기능유형의 정의

기능점수모형의 골격이 되는 시스템의 기능유형에 대한 그동안의 연구 결과를 보면 측정하는 기능유형의 종류가 <표 4>에서 보듯이 5~10개까지 매우 다양하다. 기능점수를 산출하기 위해 측정하는 기능유형의 종류가 많을수록 규모예측의 정확도는 향상되지만, 기능점수모형이 시스템 설계 이전의 개발 초기 단계에서 사용되는 점을 고려하여 활용성을 높이기 위해서는 이를 너무 세분화하지 않는 것이 바람직하다.

<표 4> 여러 기능점수모형의 기능유형  
<Table 4> Function types by models

기능점수 모형	기능유형수	기능 난이도 평가		
		단계	가중치범위	적용효과
Albrecht 모형	5	3	3~15	22~44(2배)
IFPUG 모형	5	3	3~15	24~48(2배)
Symons Mark 모형	3		0.38~1.67	
Jones 특성 점수 모형	6		3~7	
COSDES 모형	8	3	2~19	26~88(3.4배)
S/W개발비 산정기준 (정보통신부)	10	3	2~19	34~105(3.1배)

따라서 다른 모형을 개선하여 제안모형에서는 <표 5>와 같이 7개의 기능유형을 정의하여 활용도를 높이도록 하였다. 즉 화면의 경우 입력과 출력을 별도로 구현하는 경우는 거의 없기 때문에 입출력처리 화면, 메뉴화면, 그리고 조회화면으로 분류하였다. 또한 외부시스템이나 별도의 매체를 통해 시스템에 입출력되는 기능은 기타입출력처리로 정의하였다.

〈표 5〉 제안모형의 기능유형  
 〈Table 5〉 Suggested function types

기능 유형		내 용
화면	메뉴	DB 접근이 없는 입출력 화면
	조회	DB 접근이 있으나 갱신이 없는 입출력 화면
	입출력처리	DB 접근이 있으며 갱신이 수반되는 입출력 화면
출력보고서		보고서 출력
기타 입출력처리		외부 타시스템, 다른 매체 등으로부터의 입출력
DB	내부 DB	내부 화일이나 데이터베이스
	타시스템 연계용 DB	타시스템 연계용 화일이나 데이터베이스

(2) 기능유형별 기능수 측정 방법

제안모형에서 기능유형별로 기능수를 측정할 때의 기준은 다음과 같다. 특히 Client/Server환경을 고려해야 하며 또한 각 기능유형별로 중복되지 않도록 한다.

① 메뉴화면

입출력화면 중 데이터베이스 접근이 전혀 없는 화면을 메뉴화면으로 계수한다. 특히 사용자인터페이스 화면(GUI)을 Client에서 처리하고, 항목 값의 범위, 항목간 상관관계 등의 검사 기능을 Server에서 분산하여 수행토록 하는 유형이 있는데 이러한 기능은 1개의 입출력처리 기능을 분산하여 수행토록 했기 때문에 Client에서는 데이터베이스 접근 유무에 따라 메뉴화면 또는 조회화면으로, Server에서는 데이터베이스 갱신 유무에 의해 조회화면 또는 입출력화면으로 각각 계수한다.

② 조회화면

입출력화면 중 데이터베이스 접근이 있으나 갱신은 발생하지 않는 출력처리 전용 화면을 조회화면으로 계수한다.

③ 입출력화면

입출력화면 중 데이터베이스 갱신이 발생하는 입출력처리 전용 화면을 입출력화면으로 계수한다.

④ 출력보고서

데이터베이스의 내용을 프린터로 출력하는 보고서를 출력보고서로 계수한다. 보고서의 형식이 다르면 별도의 보고서로 계수하며, 화면의 내용을 그대로 하

드카피하는 보고서는 제외한다.

⑤ 기타 입출력처리

입출력 중 화면을 통하지 않고 외부 시스템과 직접 입출력 데이터를 주고받거나, 별도의 보조기억장치에 수록한 화일을 이용하여 입출력하는 경우를 계수한다.

⑥ 시스템 내부 데이터베이스 또는 화일

외부에서의 입출력 처리를 위해 내부에 유지 관리되고 있는 데이터베이스 또는 화일을 계수한다. Utility를 사용하여 관리되는 Backup화일 등은 시스템 내부 데이터베이스나 타시스템 연계용 데이터베이스로 계수하지 않도록 한다. 왜냐하면 Backup 화일을 내부 논리화일로 취급하거나, 외부출력으로 취급함으로써 기존의 기능점수와 비교하여 각각 평균 29.7%, 17.7% 정도 증가 요인이 된다는 것이 연구 결과로 밝혀진 바 있다.[5]

⑦ 타시스템 연계용 데이터베이스 또는 화일

시스템에서 최초로 생성하는 데이터베이스 또는 화일로서 타시스템에서 데이터를 입력으로 사용하는 경우에 계수한다. Client/Server환경에서 다른 컴퓨터에 동일한 데이터를 중복으로 분산하여 처리에 사용하는 경우에 원본은 시스템 내부 데이터베이스로, 동일 유형의 사본 중의 하나를 타시스템 연계용 데이터베이스로 간주한다.

(3) 기능유형별 가중치 적용

제안모형에서 기능유형별 가중치는 3단계로 구분

〈표 6〉 제안모형의 기능유형별 가중치  
 〈Table 6〉 Suggested weights on function types

기능 유형	가중치 구성비	단계별 가중치				
		단순	보통	복잡	합계	
화면	메뉴	0.066	2	3	5	10
	조회	0.120	4	6	9	19
	입출력처리	0.184	5	9	14	28
출력보고서		0.125	3	6	10	19
기타 입출력처리		0.115	3	5	9	17
DB	내부 DB	0.207	7	10	15	32
	타시스템 연계용DB	0.183	5	9	13	27
합 계		1.000				152

한다. 시스템 전체의 가중치를 1로 할 때 각 기능유형의 가중치는 <표 6>에 표시된 바와 같으며, 이 값은 여러차례의 모의실험을 통해 결정되었다. 또한 가중치의 3단계 판정 기준은 <표 7>에 나타난 바와 같다.

<표 7> 기능유형별 가중치 판정 기준  
<Table 7> Weights evaluation criteria

기능유형		판정 기준	
화면	메뉴	단순	1차 선택 조작
		보통	2차 선택 조작
		복잡	3차 이상의 선택 조작
	조회	단순	자료형 1-4, 화일수 1
		보통	자료형 5-10, 화일수 2
		복잡	자료형 11이상, 화일수 3이상
	입출력처리	단순	자료형 1-4, 화일수 1
		보통	자료형 5-10, 화일수 2
		복잡	자료형 11이상, 화일수 3이상
출력보고서	단순	화일수 1, 단순한 양식	
	보통	화일수 2, 보통 양식	
	복잡	화일수 3, 복잡한 양식	
기타 입출력 처리	단순	자료형 1-4, 화일수 1	
	보통	자료형 5-10, 화일수 2	
	복잡	자료형 11이상, 화일수 3이상	
DB	내부 DB	단순	테이블 1개, 1계층
		보통	테이블 2개, 2계층
		복잡	테이블 3개 이상, 3계층 이상
	타시스템 연계용DB	단순	테이블 1개, 1계층
		보통	테이블 2개, 2계층
		복잡	테이블 3개 이상, 3계층 이상

4.2.2 기술적 복잡도에 관한 개선

(1) 기술적 복잡도 평가 항목의 수

기존의 기능점수모형에서는 기술적 복잡도에 대한 평가항목수가 <표 8>에서와 같이 14~23개까지 정의되고 있다. 그러나 이 항목들은 대부분 중복되거나 명확한 판정기준이 난해한 상태로 정의되어 있다. 이것은 기능점수모형의 활용도를 저하시키는 원인이 된다.

<표 8> 기능점수모형별 기술적 복잡도  
<Table 8> TCF by models

구분	항목수	보정내용 및 보정식	
Albrecht 모형	14	내용	· 데이터통신의 필요 정도 등 14항목
		식	$0.01 * 14 * 5 = 0.7$
		효과	$\pm 35\%$
Symons Mark II 모형	20	내용	· Albrecht 모형에서 제시된 항목 적용 · 문서화 요구 등 6개 항목 추가
		식	$0.005 * 20 * 5 = 0.5$
		효과	$\pm 25\%$
S/W 개발비 산정기준 (정통부)	23	내용	· Albrecht 모형에서 제시된 항목 중 13개 항목 적용 · 품질 특성(7개) 및 성능 특성(3개) 항목 추가
		식	$0.006 * 23 * 5 = 0.69$
		효과	$\pm 34.5\%$

따라서 제안모형에서는 평가항목의 수를 대폭 축소하되 객관적인 기준들이 개발될 때까지는 포괄적인 방법으로 평가하기 위하여 <표 9>에서와 같이 일반적인 소프트웨어의 기술성 평가항목[13]과 Client/Server환경에서의 애플리케이션이 영향을 받는 분산 정도 등 총 5개 평가항목을 설정했다.

<표 9> 제안모형의 기술적 복잡도 평가항목  
<Table 9> Suggested TCF elements

평가 항목	평가 내용
기술	· 요구되는 기술의 복잡성 · 운영 환경의 안정성에 대한 검증정도 · 시험환경의 구축 여부 · 제반 자원의 연결 필요성
성능	· 처리속도, 처리율, 응답시간, 반응시간
품질	· 정확성, 보안유지성, 효율성, 변경용이성, 무결성, 신뢰성, 이식성
사용 용이성	· 조작편리성, 습득용이성, 설치용이성, 백업, 온라인도움말, 인간공학성
분산 정도	· 프로세스의 분산 정도 · 데이터의 분산 정도

(2) 기술적 복잡도 평가 결과의 적용

제안모형에서 기술적 복잡도에 대한 평가항목별로



프로젝트에서 전혀 고려할 필요가 없는 경우를 0으로 하고 고도의 엄격한 요구사항을 만족해야 하는 경우를 5로서 각각 평가하여 합산한다. 이와 같이 평가된 기술적 복잡도에 의한 영향도는 0~25의 범위에 들기 때문에 총영향도(TDI)를 (식 6)에 의해 ±35%로 보정하고, 총기능점수는 (식 7)에 의해 산출한다.

$$TCF = 0.65 + (0.035 * TDI) \quad (6)$$

$$\text{총기능점수}(FP) = \text{기본기능점수}(FPu) * TCF \quad (7)$$

### 4.3 제안모형에 대한 평가

#### 4.3.1 평가 방법

실제 수행된 프로젝트에 제안모형을 적용하여 소프트웨어규모 예측능력을 프로젝트별 실무인력과 회귀식으로 평가하고, 이 결과를 다른 기능점수모형에 의한 규모예측의 결과와 비교하여 제안모형의 정확도를 평가한다.

#### (1) 실적 자료 수집 및 분석

실적자료는 96년 현재를 기준으로 국영기업체의 SI전문회사인 K사에서 최근 2년동안 수행한 Client/Server환경에서 개발된 사무처리용 프로젝트를 대상으로 24건의 실적 자료를 수집하였다. 이 표본은 동일 조직과 프로젝트 유형으로 설정되어 편차가 최소화 된 것이다. 수집된 실적 자료의 내용은 다음과 같다.

#### ① 프로젝트별 실제 투입인력

한 프로젝트에 투입된 인력은 특급기술자로 부터 초급기능사까지 다양한 등급의 기술자가 포함되므로, 프로젝트별 전체 투입인력을 등가로 환산하기 위해 중급기술자를 기준하여 정부고시인건비 단가를 적용하여, 단위는 인·월(Man-Month)로 표현하였다. 인력 규모는 표준 소프트웨어개발공정에서[1] 전체 공정에 투입된 인력이 아니라 개발업체에서의 용역 분에 해당되는 시스템설계, 구현, 설치 및 시험공정에 투입된 인력이며 프로젝트 계획 및 관리, 기초요구분석, 상세요구분석, 기타 검수 공정 등 발주자 수행분에 대한 인력은 제외되었다.

#### ② 기능유형별 기능수

7개 기능유형별 기능수는 단순, 보통, 복잡의 3단계로 가중치를 적용하여 측정된 기능수이다. 제안모형

의 정확도를 검증하기 위해서 동일 프로젝트에 대해 제안모형뿐만 아니라 Albrecht모형, IFPUG모형, COSDES 모형에 의한 기능유형별 기능수도 각각 측정하였다.

#### (2)총기능점수 산정

제안모형에 의한 기능유형별 기능수로 기본기능점수를 산정하고 기술적 복잡도에 따른 보정을 거친 결과 총기능점수가 <표 10>과 같이 산출되었다.

<표 10> 제안모형의 기능점수 산정 결과  
<Table 10> Estimated function points using suggested model

No	기능수 (FC)							기본 기능 점수 (FPu)	총 기능 점수 (FP)
	화 면			보고서	기타	화일/DB			
	메뉴	조회	입출력			내부	연계		
P1	13	0	129	23	24	40	61	2,470	2,643
P2	12	39	123	17	19	10	43	2,364	2,529
P3	12	13	108	29	21	42	57	2,126	2,275
P4	11	0	190	88	50	64	0	3,301	3,301
P5	2	0	42	24	0	25	0	853	1,749
P6	9	0	65	10	2	22	0	1,022	1,022
P7	7	138	52	0	0	21	0	835	835
P8	11	9	84	75	0	38	0	960	960
P9	26	27	38	21	6	8	4	886	917
P10	43	81	75	20	32	33	4	1,900	1,967
P11	5	12	43	19	11	35	0	915	755
P12	5	14	14	9	2	2	0	263	208
P13	3	6	6	1	4	3	0	133	105
P14	2	14	7	6	0	8	0	243	209
P15	12	60	260	80	13	48	0	4,342	4,646
P16	8	8	78	142	9	32	0	2,288	1,888
P17	6	7	38	8	3	11	0	513	513
P18	4	15	65	15	157	33	0	1,772	1,462
P19	2	30	59	50	5	12	0	1,063	1,100
P20	2	0	2	288	5	7	0	596	492
P21	4	100	45	90	0	40	0	1,820	1,947
P22	7	7	84	15	3	15	3	1,310	1,035
P23	6	6	103	37	0	20	2	1,631	1,346
P24	24	49	87	496	0	106	0	4,017	4,298

#### (3)기능점수모형별 총기능점수 산정

제안모형과 기존 기능점수모형의 정확도를 비교하기 위해 각 모형별 총기능점수를 측정 한 결과는 <표 11>과 같다.

#### 4.3.2 평가 결과 및 분석

기능점수모형의 장점 중의 하나는 총기능점수를 프로젝트의 생산성, 품질, 소요 비용의 기준으로 활용

〈표 11〉 모형별 기능점수 산정 결과  
 〈Table 11〉 Estimated FP by models

No	투입인력 (MM)	총 기능점수 (FP)			
		제안모형	Albrecht	IFPUG	COSDES
P1	75	2,643	2,363	2,433	2,206
P2	72	2,529	2,312	2,486	2,549
P3	60	2,275	2,354	2,437	1,840
P4	129	3,301	3,085	3,387	3,265
P5	23	1,749	866	926	910
P6	35	1,022	1,081	1,111	1,213
P7	41	835	990	1,285	785
P8	30	960	1,183	1,363	892
P9	27	917	724	853	863
P10	80	1,967	1,760	2,090	2,086
P11	17	755	966	1,061	799
P12	11	208	256	321	280
P13	10	105	132	150	118
P14	8	209	213	277	246
P15	127	4,646	4,058	5,117	5,819
P16	64	1,888	2,508	3,345	2,917
P17	19	513	448	482	548
P18	62	1,462	1,433	1,561	1,508
P19	30	1,100	1,057	1,255	1,143
P20	21	492	498	748	539
P21	58	1,947	1,875	2,477	1,775
P22	38	1,035	1,372	1,435	1,649
P23	47	1,346	1,617	1,801	1,859
P24	119	4,298	6,985	5,760	3,982

할 수 있다는 것이다. 따라서 본 논문에서 제안된 모형의 정확도를 검증하기 위해서 프로젝트별 총기능점수와 투입인력과의 관계를 회귀모형으로 정의해 보고자 한다. 일반적으로 총소요인월을 예측하기 위한 계산식의 회귀모형은 다음 (식 8)과 같이 정의된다.[14]

$$\text{총소요인월}(MM) = b_0 * \text{총기능점수}(FP)^{b_1} \quad (8)$$

여기서  $b_0$ ,  $b_1$ 은 계수이다. 〈표 10〉의 총기능점수는 제안모형에 의해 산정된 것이며, 〈표 11〉의 투입인력은 실제 데이터이다. 통계분석결과 제안모형의 (식 8)에 적용될 계수는  $b_0 = 0.075$ ,  $b_1 = 0.885$ 로 계산되었으며 ( $F = 308 > f_{2,22, \alpha=0.05} = 3.89$ ), 따라서 (식 8)을 다시 쓰면 다음과 같다.

$$MM = 0.075 * FP^{0.885} \quad (9)$$

〈표 12〉에서 볼 수 있듯이 본 모형에 대한 회귀모형

의 결정계수( $R^2$ )는 0.965로 나타났으며, 기존의 Albrecht, IFPUG, COSDES 모형 등 다른 기능점수모형에 의해 실시한 회귀분석 결과보다 우수한 수준의 것임을 알 수 있었다. 이 결과는 제안모형이 기존의 기능점수모형에 비해 개선된 것임을 보여준다.

〈표 12〉 회귀 분석 결과  
 〈Table 12〉 The results of regression

모형	$b_0$	$b_1$	$R^2$
제안 모형	0.075	0.885	0.965
Albrecht 모형	0.324	0.691	0.939
IFPUG 모형	0.109	0.822	0.955
COSDES 모형	0.138	0.803	0.959

### 5. 결론

본 논문에서는 기존의 기능점수모형을 개선하여 활용성과 정확도를 높이고 특히 Client/Server환경에도 적용할 수 있는 모형을 제안하였다. 즉 프로젝트에 대해서 7개의 기능유형별 기능수를 측정하고, 5개의 기술적 복잡도 평가항목에 의해 평가된 영향도를 이용해 산출된 기능점수를 이용하여 소프트웨어의 규모를 예측하는 모형이다. 이 모형은 다른 기능점수 모형에서 보다 기능유형과 기술적 복잡도 평가항목이 축소되어 활용하기에 용이하다. 또한 이 모형의 정확도는 기완료된 프로젝트 24건의 자료를 적용한 결과 기존의 기능점수모형보다 우수한 수준으로 나타났다.

이 모형을 보다 개선된 기능점수모형으로 발전시키기 위해서는 많은 실적자료를 통해 가중치의 객관성을 높여나가야 하겠다.

### 참고 문헌

- [1] "소프트웨어개발비용 산정기준 및 해설", 과학기술처, 1991. 1.
- [2] A. J. Albrecht and J. E. Gaffney, "Software Function, Source Line of Code, and Development Effort Prediction: A Software Science Validation," IEEE TSE Vol. 9, No. 6, pp. 639-647, 1983. 11.

[3] B. Boehm, "Software Engineering Economics," Englewood Cliffs, NJ, Prentice-Hall, 1981.

[4] C. R. Symons, "Function Point Analysis: Difficulties and Improvements," IEEE TSE, Vol. 14, No. 1, pp. 2-11, 1988.

[5] C. F. Kemerer and B. S. Porter, "Improving the Reliability of Function Point Measurement: An Empirical Study," IEEE TSE Vol. 16, No. 1, pp. 64-71, 1990.

[6] G. C. Low and R. Jeffery, "Function Points in the Estimation and Evaluation of the Software Process," IEEE TSE, Vol. 16, No. 1, pp. 64-71, 1990.

[7] C. Johns, "Programming Productivity," New York : McGraw-hill, 1986.

[8] T. Mukhopadhyay and S. Kekre, "Software Effort Models for Early Estimation of Process Control Applications," IEEE TSE, Vol. 18, No. 10, pp. 915-924, 1992.

[9] 田中淳, "Client/Server의 어려운 문제-개발공수의 견적에 도전", 일경컴퓨터, pp. 114-125, 95.3.20

[10] 정보통신부, "소프트웨어개발비 산정기준의 문제점과 개선방향", 소프트웨어산업협회보, 제25호, pp. 27-29, 1996.

[11] 심기보, "일본의 클라이언트/서버 비즈니스의 견적방법 소개", 소프트웨어산업협회보, 제29호, pp. 31-35, 1997.

[12] C. R. Symons, "Function Point Analysis: Difficulties and Improvements," IEEE TSE Vol. 14, No. 1, pp. 2-11, 1988.

[13] 과학기술정책관리연구소, "소프트웨어의 기술성 평가방안에 관한 연구", 한소협, 1993. 11.

[14] 이주현, "실용 소프트웨어 공학론", 법영사, pp. 131-133, 1993.



안 연 식

1982년 전북대학교 자연과학대학 졸업(이학사)  
 1990년 연세대학교 산업대학원 전자계산전공(공학석사)  
 1997년~현재 국민대학교 대학원 정보관리학과 박사과정 재학중

1981년~1992년 한국전력공사 정보처리처 과장  
 1992년~1996년 한전정보네트웍(주) 컨설팅사업부장  
 1996년~현재 경원전문대학 교수(정보처리기술사)  
 관심분야: OA, MIS, S/W공학(방법론), 분산DB