DOI: 10.3745/KIPSTD.2008.15-D.3.375

객체지향 기반 효율적인 기능점수 측정 프로세스 설계 및 사례연구

김 동 선[†] · 윤 희 병[††]

요 약

최근 정보시스템 개발 패러다임이 객체지향 및 컴포넌트 기반으로 변해가고 있으며 이러한 방법론이 소프트웨어 산업계를 주도하고 있다. 소프트웨어 비용측정 분야도 이러한 변화에 적응하기 위해서 OO 환경에 적합한 비용 측정 모델로의 전환이 요구되고 있으며 실제 UCP 기법 등 객체지향 개념의 기능점수 연구가 진행되고 있다. 특히 객체지향 기능점수 측정 기법은 사용사례 중심으로 개발 프로젝트 초기에 적용 가능하고, 개발 진행 단계별 소프트웨어 산출물에 따라 생명주기 전체에 걸쳐서 적용 가능하고, UML 표기법에 의한 사용자와 의사소통으로 이해가 항상 가능하기 때문에 LOC 및 기존 기능점수 측정 기법의 한계를 극복하는 장점이 있다. 따라서 본 논문은 최근 객체지향 정보시스템 개발 환경에서 널리 사용되고 있는 UML 표기법 및 국방 CBD 방법론 절차에 의한 정보시스템 개발 프로젝트 진행에 따라 초기단계 시점(ROFP)과 분석단계 시점(AOFP)에서 기능점수를 측정하며, 전통적인 FPA 모델과 객체지향 FPA 모델을 통해 UML 모델링과 산출물을 기반으로 한 유스케이스와 클래스의 상관도 분석에서 서비스 기능과 객체/클래스 기능을 식별하고 기능점수를 구하는 측정기법을 제안한다. 무엇보다 전통적인 기능점수 측정 기법인 IFPUG-CPM 및 소프트웨어 대가기준의 기능점수 측정기법을 개선하여 객체지향 정보시스템 개발에 적합한 기능점수 측정 프로세스를 제시하고 사례연구를 통한 적합성 평가 결과를 제시한다.

키워드 : 기능점수 측정, 객체지향 기능점수, UML 상관관계 분석, 기능점수 측정 프로세스

Process Design and Case Study for Efficient Function Point Measurement Based on Object Oriented

Dongsun Kim[†] · Heebyung Yoon[††]

ABSTRACT

Recently, development paradigm of information system is turning into object oriented and component based, and this methodology is leading the software industry. To acclimatize aptly to this trend, users demand the assessment of software expenses to change with the appropriate model of computing costs of the environment, and some people are actually studying the concept of Object Oriented Function Point and UCP method. Especially, Object Oriented Function Point Measurement Process has good points in overcoming the bound of LOC and the existing the Function Point Measurement Process because Object Oriented Function Point Measurement Process is applicable to the early stage of development project mainly with the used cases, and valid to the life long period as the each stage of software products develops, and always understandable to communicate with users by the UML mark rules. Accordingly, this research is to measure Functional Point at ROFP and AOFP in accordance with the development project of information system by the national defense CBD methodology procedures and UML Interrelation Analysis that are recently and widely used in the developmental environment of object oriented information system. Furthermore, this study suggests the measurement method to obtain Functional Point, and identifies service function and object/class function in the correlation analysis of use case and class based on the products and UML modeling via traditional FPA model and object oriented FPA model. Above all, this study is to demonstrate the improvement of traditional Function Point Measurement Process, IFPUG-CPM and software cost basis, and reveal Function Point Measurement Process, which is appropriate to the development of object oriented information system, and suggest the evaluation results of the compatibility through case studies.

Key Words : Function Point Measurement, Object Oriented Function Point, UML Interrelation Analysis Function Point Measurement Process

1. 서 론

정보시스템 개발비 산정방식에는 개발 규모에 의한 산정방식과 투입인력수와 기간(man/month)에 의한 산정 방식이 있으며 개발 규모를 측정하는 기법으로는 코드 라인 수(LOC)

† 준 회 원 : 국방대학교 전산정보학과 석사과정
†† 종신회원 : 국방대학교 전산정보학과 교수
논문접수 : 2008년 2월 21일
수 정 일 : 2008년 4월 11일
심사완료 : 2008년 4월 15일

방식과 기능점수(FP: Function Point) 방식이 있다. 한국은 주로 정보시스템 개발 사업의 LOC 방식을 사용해 왔는데 개발자 중심으로 이해하기 쉽고 의미가 명확하여 지금까지 널리 사용되어 왔으나, 프로젝트 초기에 라인 수 예측이 어렵고 개발 언어나 시스템 환경 요소에 영향을 받으며, 또한 잘 작성된 프로그램이 오히려 비용을 적게 받는 경우가 발생하기도 한다. 그런 측면에서 규모 측정 방법의 근거가 불명확하고 측정 시점에 따라 하자가 발생하는 등 여러 가지 문제점이 있다. 기능점수 방식은 사용자 관점에서 업무적 요구기능을 정량화하여 규모를 측정하기 때문에 사용자가 이해하기 쉽고 개발 언어나 환경에 독립적이며 사업 계획단계에서도 사업규모 측정이 가능하다[1].

이러한 추세에 따라 기능점수 측정을 위한 활동이 활발하게 일어나고 있으며 ISO에서는 세계 여러 단체에서 제시하는 기능점수 측정 방법이 국제 표준에 부합하는지 평가하여 부합되는 것만 국제 표준으로 인정하고 있다. 이렇게 인정된 모델은 ISO/IEC 20926인 IFPUG-CPM, ISO/IEC 19761인 COSMIC-FFP, ISO/IEC 20968인 Mark Ⅱ, ISO/IEC 24570인 NESMA 등 4가지 종류가 있으며[1], 정통부에서는 IFPUG-CPM을 표준으로 채택하고 있다.

이러한 FPA(Function Point Analysis) 모델들은 전통적인 개발환경에서는 대중적이며 양질의 예측방법이지만 최근 정보시스템 개발이 객체지향으로 패러다임이 변해 있어 OO(Object Oriented) 개념 및 특성이 FP에 적용되지 않으면 FPA를 하는데 상대적으로 어렵다. 최근 CBD의 등장으로 OO 환경에서 기능점수 측정의 요구가 나타나고 있으며 이러한 접근은 최종 측정 결과보다는 착수단계에서 부터 점진적인 개발을 진행하며 개발 규모를 산정하는데 있다[2]. 하지만 구조적 개발 방식에 기반을 두고 있는 FPA 모델을 측정시 전문화된 기술수준을 요구하고 있고 UML이나 CBD에 직접적으로 적용하는데 여러 가지 어려움이 존재하고 있다[3][4].

따라서 본 논문은 OO 기반 정보시스템 개발의 요구사항 도출 및 요구분석 단계 산출물을 이용한 객체지향 기능점수 측정 프로세스를 제시한다. 이를 위해 전통적인 FPA 사고방식을 OO 개념 모델에 적용시켜 유스케이스와 클래스의 상관도 분석으로 서비스 기능 및 클래스 기능을 식별하고 기능점수를 계산하는 프로세스를 설계한다. 마지막으로 사례 연구를 통해 프로젝트의 착수단계(ROFP: Require Object Function Point)와 분석단계(AOFP: Analysis Object Function Point)에서 객체지향 기능점수를 측정하고 제안한 기법의 적합성을 평가한 후 그 결과를 제시한다.

2. 관련 연구

2.1 기능점수 측정의 표준 모델 고찰

기능규모 측정(FSM: Functional Size Measurement)은 ISO에서 1998년 ISO/IEC 14143을 표준으로 사용자에 의해 요구되는 기능들에 관련된 크기를 측정하기 위해 초점이 맞추어져 있다[5]. 이렇게 ISO/IEC 14143에 의해 인정된 모델은 ISO/IEC 20926, 20968, 24570, 19761 등 4종류가 있으며, 정보통신부 소프트웨어 사업대가 기준은 ISO/IEC 20926을 따르고 있다.

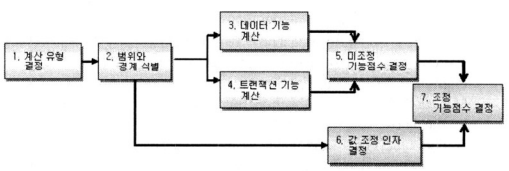
따라서 본 연구에서는 IFPUG-CPM 모델 (ISO/IEC 20926)의 고찰을 통하여 새로운 객체지향 기반 측정 프로세스 설계를 위한 측정 개념을 연구한다.

기능점수 측정 규칙들은 1979년 IBM의 Albrecht에 의해 최초로 소개된 이래 계속 검토되고 명료화되어 더욱 일관성을 갖도록 지속적으로 다듬어져 왔다. IFPUG에서는 계산 실무 매뉴얼을 발표하였는데, 여기에는 측정 과정에 관한 최신의 표준과 지침이 포함되어 있다[14].

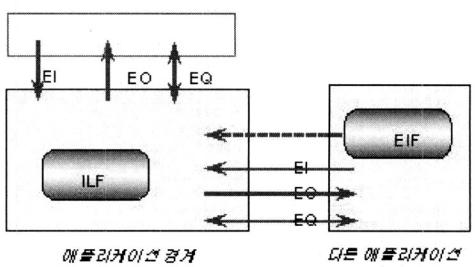
기능점수 분석은 소프트웨어 개발과 유지보수 프로세스 전반에 걸쳐 사용자들에게 제공되는 애플리케이션 기능의 양을 측정하는데 사용되는 전체 프로세스는 (그림 1)과 같이 7단계로 요약할 수 있다[7].

1단계는 기능점수 계산 유형을 결정한다. 유형에는 개발 프로젝트 기능점수(DFP: Development Function Point), 유지보수 프로젝트 기능점수(EFP: Enhancement Function Point), 애플리케이션 기능점수(AFP: Application Function Point)의 3가지 유형이 있다. 2단계는 계산 범위와 애플리케이션 경계를 (그림 2)와 같이 식별한다.

3단계는 모든 데이터 기능들과 그 복잡도를 식별한다. 사용자에게 제공하는 데이터유형으로 (그림 2)와 같이 식별된 내부논리파일(ILF: Internal Logical File)과 외부 인터페이스(EIF: External Interface File)에 대해 데이터 요소 유형(DET: Data Element Type)과 레코드 요소 유형(RET: Record Element Type)의 기능 복잡도를 할당하고 기여도를 계산한다. 4단계는 모든 트랜잭션 기능들과 그 복잡도를 식별한다. 트랜잭션 기능은 3종류가 있는데 외부입력(EI: External Input), 외부출력(EO: External Output), 외부조회(EQ: External



(그림 1) 기능점수 측정 절차



(그림 2) 기능점수 측정 구성요소

Query)가 있으며, 참조파일 유형(FTR: File Types Referenced)과 DET의 개수로 복잡도를 결정하고 기준에 따라 기여도를 계산한다. 5단계는 데이터 기능점수와 트랜잭션 기능점수를 합하여 미조정 기능점수(UFP: Unadjusted Function Point) 값을 결정한다. 6단계에서는 14개의 일반 시스템 특성(GSC: General System Characteristics)에 대해 0에서 5까지 영향도(DI: Degree of Influence)를 평가하여 총 영향도(TDI: Total Degree of Influence)를 계산하고 미조정기능점수 값을 보정한 (TDI * 0.01) + 0.65를 계산하여 값 조정인자(VAF: Value Adjustment Factor)를 결정하며, 마지막 7단계는 최종적으로 UFP에 VAF를 곱하여 조정 기능점수(AFP: Adjusted Function Point) 값을 계산한다.

2.2 객체지향 분석 프로세스

개발 패러다임이 객체지향으로 변해 감에 따라 기존 기능점수 측정 기법을 그대로 적용하기에 제약사항이 있어 이를 해결하게 위한 객체지향 기능점수 측정 기법들이 활발하게 연구되고 있다. 본 절에서는 정보시스템 개발 초기 단계에서 객체지향 기능점수 규모측정에 목적을 두고 있어 요구분석 단계의 핵심 프로세스에 대해서 분석한다. 객체지향 개발은 다음 (그림 3)과 같이 9단계의 프로세스를 수행한다.

객체지향 개발의 생명주기는 계획, 분석, 설계, 구현, 테스팅의 반복과정을 거친다. 이러한 프로세스에서 분석단계의 핵심 3단계에 대해 기술하면 다음과 같다.

첫 번째 단계인 요구사항 정의는 구축되어야 할 정보시스템의 요구사항을 파악하기 위해 개발 시스템에 대한 사용자의 기능적 및 비기능적 요구사항을 식별하여 요구사항명세서를 작성한다. 두 번째 단계는 기능적 모델링으로 유스케이스 다이어그램을 사용하여 시스템의 동적인 관점에서 모델링하고, 일반화, 포함, 확장 관계로 유스케이스 조직을 분류할 수 있다. 세 번째 단계는 객체모델링으로서 클래스 다이어그램을 사용하여 시스템을 정적인 관점에서 가시화, 명세화, 문서화하기 위해 사용된다. 클래스 간에 단순한 연관관계를 가질 수도 있지만 복합관계와 집합관계, 일반화관계, 의존관계로 표현될 수 있다[6].

2.3 선행 연구된 객체지향 기능점수 기법

2.3.1 UCP 측정 기법

1993년 Karner의 연구에 따르면 UCP는 FPA와 MarkⅡ 기능점수의 확장으로서 사용자 관점에서 기능성이 있다는 것에 기본을 두고 있다, 이 방법은 유스케이스 수를 기반으로 소프트웨어 크기를 예측하는 방법으로 초기에 예측을 위해서는 문제 도메인, 시스템 크기 및 아키텍처를 이해하고 있어야 한다[4].

UCP는 식별된 액터와 유스케이스를 유형에 따라 분류하고 조정인자의 적용 여부에 따라 미조정 유스케이스 점수(UUCP: Unadjusted Use Case Point)와 조정 유스케이스 점수(AUCP: Adjusted Use Case Point)로 분류된다. UUCP를 측정하기 위해서는 각 액터의 복잡도에 대한 가중치를 합하여 미조정 액터 가중치(UAW: Unadjusted Actor Weights)를 산출하고, 여기에 각 유스케이스 복잡도 가중치를 합하여 산출된 미조정 유스케이스 가중치(UUCW: Unadjusted Use Case Weights)를 합산하여 측정한다.

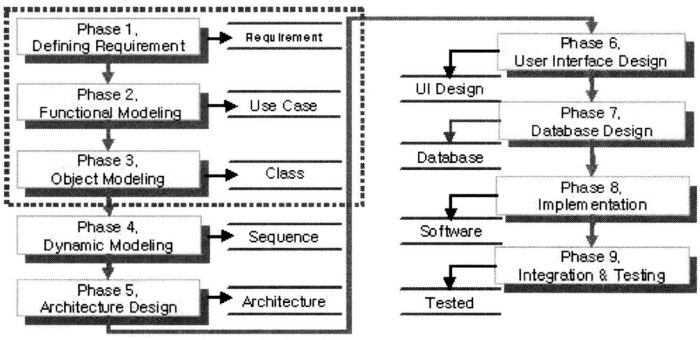
2.3.2 OO-Jacobson 접근을 통한 FPA에 사상 기법

1997년 Fetcke는 FPA에 객체지향 접근방법을 적용시켜 소프트웨어의 기능적 크기를 측정하기 위해서 측정 프로세스를 지원하는 간결한 규칙의 집합을 바탕으로 기능점수 모델에 유스케이스 모델을 직접적으로 적용시키는 방법을 제안했다[3].

이 기법은 기능점수 분석 구현에 사용되는 기술에 독립적이며, IFPUG-CPM의 기능점수 분석을 객체지향 프로세스의 분석단계에서 결과물에 적용하여 기능점수를 측정하는데 초점을 두고 있다. 즉 OOSE로 개발된 소프트웨어 애플리케이션에 기능점수 측정을 위한 계산 규칙을 적용하기 위하여 FPA와 OOSE를 분석하여 구체적인 사상 규칙을 제시하였다.

2.3.3 시퀀스 모델을 이용한 기능점수 측정 기법

1999년 Uemura는 미조정 기능점수의 계산에 주안점을 둔 상세 기능점수 측정에 관한 연구를 목적으로 UML에 기반한 요구사항과 설계명세에 IFPUG 버전의 5단계를 적용시키고, 데이터 기능 유형을 계산하는 규칙과 트랜잭션 기능 유형을 계산하는 규칙을 UML의 시퀀스 다이어그램에 입각하여 정



(그림 3) UML 기반 분석·설계 프로세스

의했다[8].

이를 위해서 트랜잭션 기능 유형을 계산하는 규칙을 정의할 때는 설계단계에서 도출된 시퀀스 다이어그램을 이용하고 기능점수 분석단계는 전통적인 기능점수 분석 단계와 동일하나 각 단계마다 적용된 규칙은 객체지향 특성에 맞게 변형되었다.

2.3.4 기타 객체지향 기능점수 관련 선행 연구

NUC(Normalized Use Case)는 개선된 유스케이스 점수산정 방식을 위해 2002년 발표된 방법론으로 산정방식이 공개되지 않는 산정 시스템이 제공되고 있다. 이후 미국 정부의 지원하에 Galorath Inc.에 의해 계속적으로 연구되고 있는 방법론이다[6].

또한 Donald는 LOC나 전통적인 기능점수 기법으로는 멀티미디어 파일, 스크립트, 웹 빌딩 블록(Active X, Applets, OLE 등), 링크(XML, HTML 등)를 측정하기 어렵다는 단점을 지적하고 전통적인 기법에 추가적으로 웹 객체들에 대해 기능점수를 계산하는 기법을 제안했다[9].

3. 기능점수 측정 현황 및 분석

3.1 기능점수 측정 현황

3.1.1 국내 현황

소프트웨어 사업대가기준은 2004년에 ISO/IEC 20926 (IFPUG-CPM)의 소프트웨어 규모 산정기준인 기능점수와 소프트웨어 생명주기의 개발공정(ISO12207)을 도입하여 규모 산정 및 적용 공정의 정확성을 제고했다. 기존 본수 방식을 폐지하고 LOC 방식과 투입인력 및 기간에 의한 방식(man/month)은 이전 방식을 유지했으며, 보정계수 체계를 현실화하여 정보기술의 발전과 사회환경의 변화에 맞게 조정하여 현재 정통부고시 제2007-20호에서는 기능점수 및 LOC 방식을 일부 개정했다[10].

소프트웨어진흥원에 의하면 2005년 이후부터 소프트웨어 개발비 산정 방식의 활용 추세는 기능점수와 man/month 방식을 많이 사용하고 있지만 향후 추세는 기능점수에 의한 방식을 사용할 것이다[11]. 또한 소프트웨어 개발규모의 정량화가 가능한 국제표준 채택으로 해외시장 진출의 발판 마련을 위해 기능점수 방식에 의한 사업발주를 권장하고, 기획예산처에서는 2005년부터 예산 심의시 소프트웨어 개발비는 기능점수에 의해 산정할 것을 권장하고 있다.

3.1.2 국외 현황

미국, 영국, 독일 등 선진국은 소프트웨어 사업 등의 공정한 경쟁체계 확립을 위해 다양한 지원활동을 실시하고 있다. 이러한 목표를 달성하기 위해 연방발주관리개선법(FARA), 정보기술관리개선법(CCA) 등 각종 법률 및 제도를 확립하고 있다. ISO/IEC에서는 기능점수 방식과 관련된 다양한 표준들이 연구·제정되고, 이미 소프트웨어 비용 산정을 위한 다양한 모형과 도구들이 개발되어 사용되고 있다. 2001년에 CMMI(Capability Maturity Model Institute) 및 NASA에서 규모 산정 표준으로 기능점수를 도입했으며 국제기구에서는 2002년에 IFPUG-CPM 4.1을 국제표준으로 채택했다. 또한 미국 국방부(DoD)에서는 합리적인 비용산정모형 사용을 규정하고 있고 호주에서는 2000년에 빅토리아주를 중심으로 기능점수제도를 도입했으며 일본에서는 1994년에 JFPUG(일본 기능점수 사용자그룹)를 발족하여 2003년부터 ISO/IEC 14143-6을 주도적으로 개발하고 있다[11].

아시아권 국가에서에서는 처음으로 정보화 사업 측정기준을 조사 분석하고 해외사례를 비교하여 합리적인 측정지표 발굴을 위해 KFPUG, JFPUG, CSBSG(중국 소프트웨어 벤치마킹 표준그룹)가 한·중·일 소프트웨어 메트릭 연합을 결성하여 운용하기로 하는 등 활발하게 활동이 이루어지고 있다.

3.2 기능점수 측정기법 분석

3.2.1 전통적인 기능점수 측정기법 분석

IFPUG에서는 전통적인 기능점수를 발전시켰고 이후 이를 보완하기 위해 MarkⅡ 기법을 객체지향 계산에 적합하도록 테일러링했으며, NESMA 기법은 빠른 시점의 예측과 단순화에 중점을 두었고, FFP 기법은 내장형 소프트웨어나 다층(multi-tier)에서 계산할 수 있도록 했으나 전통적인 기존 기능점수 기법들은 모두가 구조적인 규모산정 방식을 벗어나지 못하고 있다.

이러한 전통적인 기법들은 사용자 관점에서 사용자의 요구사항을 정량화할 수 있는 기회를 제공하고 LOC의 코딩 단계 이후의 늦은 시점에서 규모 측정의 단점을 해결하여 소프트웨어 생명주기의 분석/설계 등의 이른 시점에서 수행할 수 있도록 했으나 실질적으로 기능점수 측정에 있어 전문가적인 기술수준을 요구하고 있어 어려움을 겪고 있다.

3.2.2 객체지향 기능점수 측정기법 분석

객체지향 및 구조적 방법론 모두의 요구사항을 도출하기 위해 유스케이스를 사용할 수 있으며 Karner[4]의 UCP 기법은 이러한 단일화된 규모 측정 기법의 전제 조건을 대부분 만족시킬 수 있고 객체지향 구현을 위한 개발 초기단계에서 활용될 수 있다는 점에서 매우 매력적인 방법으로 알려져 있다. 하지만 선행 연구된 이러한 객체지향 기능점수 측정 기법들은 직접 적용하기에 다음과 같은 문제점들로 인해 다소 무리가 있다.

첫째, UCP 측정 기법의 문제점은 트랜잭션 기능인 동적인 측면만 강조하고 있으며, 유스케이스 모델과 명세의 상세도가 다양할 수 있어 이러한 점은 UCP의 측정을 어렵게 만들고 결과의 신빙성을 감소시킬 수 있다. 또한 제시된 유스케이스 점수 당 소요 노력량 인자가 개발업체 마다 실제 프로젝트 적용결과와 차이가 발생한다는 것이다.

둘째, OO-Jacobson 접근을 통한 FPA에 사상 기법의 문제점은 데이터 기능인 정적인 측면만 강조하고 있으며, 데이터 요소의 유형을 결정하고 이를 계산하는 단계의 제안은 객체지향 개념과 전통적 기능점수 방법을 충실히 사상시킨 반면

트랜잭션 요소의 유형 결정과 계산에 대해서는 객체지향 개념의 고려가 거의 없으며 단순히 IFPUG-CPM의 참조만을 지시하고 있다는 것이다.

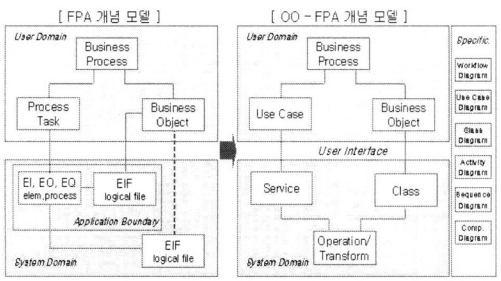
셋째, 시퀀스 모델을 이용한 기능점수 측정 기법의 문제점은 UML을 이용한 분석에 있어 중요한 모델인 유스케이스에 대한 논의가 전혀 없었으며, 제안 단계 중 데이터 요소 유형을 결정하고 계산하는 과정이 지나치게 단순화 되어 있으며 트랜잭션 유형 결정을 위해 제안된 규칙이 전통적인 FPA에서 요구하는 사항을 충실히 이행하지 못하고 있다는 것이다.

3.3 기능점수 측정의 문제점 해결 및 측정 방안

먼저 기능점수 측정 절차에 있어 UML 및 객체지향 분석 산출물을 이용하여 기능식별의 일관성을 유지하고 누구나 기능 식별이 가능한 기능점수 측정 프로세스를 설계한다. 다음으로 유스케이스와 클래스를 사용하여 정적 및 동적 요소를 고려하고 측정의 정확도와 정합성을 향상시키며, 우리 실정에 부합하는 소프트웨어 대가기준 가중치와 보정계수 사용으로 측정의 신뢰성을 높인다.

FPA 모델은 애플리케이션에서 사용자 도메인과 사용자가 요구한 사항을 어떻게 구현할 것인가에 대한 애플리케이션 도메인 영역의 결합으로 이뤄진다. 논리파일은 비즈니스 객체로 정의된 기능적 사용자 요구사항(FUR: Functional User Requirements)을 구현하고 요소 프로세스는 프로세스 타스크에 의해 정의된 FUR을 구현한다. 실질적인 FPA 개념의 접근은 (그림 3)의 왼쪽과 같다. 프로세스 타스크로 설명되는 FUR은 요소 프로세스인 EI, EO, EQ의 기능으로 구현된다. 비즈니스 오브젝트로 설명되는 FUR은 애플리케이션 영역 안의 ILF와 애플리케이션영역 밖의 EIF로 구분된다[12]. (그림 4)의 오른쪽은 OO 환경에서의 규모 측정 모델을 보여준다. 사용자 인터페이스에 의해 사용자 도메인과 시스템으로 구분하고 명세를 위한 모델과 다이어그램은 UML로 표시한다. 예외적으로 요소 프로세스는 FPA 규칙을 기반으로 일반적이고 유사한 정도의 규모 측정 패러다임을 사용한다.

OO 환경에서 클래스의 속성과 오퍼레이션은 기능점수를 계산하는 중요한 요소들이다. 즉 이러한 OO 환경의 개념 모델 접근은 유스케이스, 클래스 등 UML의 견해와 일치하고 OO 환경에서 객체지향의 기능점수 계산을 어떻게 구하는지를 설명한다.



(그림 4) FPA 개념모델과 OO-FPA 개념모델 비교

4. 기능점수 측정 프로세스 설계

4.1 객체지향 기능점수 측정의 구성

본 논문에서 제안하는 프로세스의 핵심은 FPA 사고방식을 OO 개념 모델에 적용하기 위해 사용자 도메인 부분(FUR)인 유스케이스와 시스템 도메인 부분(BFC: Base Functional Component)인 클래스를 이용하여 요소 프로세스의 기능크기 계산에 초점을 두고 있으며, 이를 바탕으로 복잡도/가중치와 보정계수를 적용한다. 규모측정은 (그림 5)와 같이 기능크기 계산, 복잡도/가중치 부여, 보정계수 적용 등 3가지로 이루어진다.

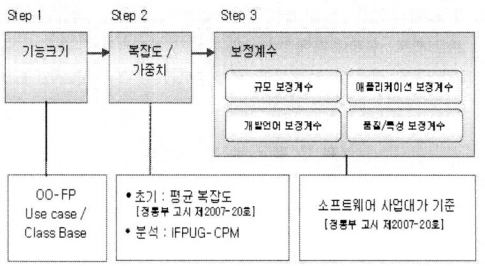
본 논문에서는 Kammelar[2]의 객체지향 애플리케이션 규모산정 구조의 원리를 응용하고 유스케이스와 클래스의 상관도 분석을 통해 이루어진다. 유스케이스 서비스와 비즈니스 객체 클래스의 유기적인 관련성 분석을 통해 트랜잭션 기능과 데이터 기능을 도출한다. 복잡도/가중치 부여에 있어 개발 초기단계와 분석단계에서 다르게 적용하며, 평균 복잡도/가중치는 소프트웨어 대가기준[13]의 평균 복잡도/가중치를 따르고, 분석단계의 복잡도/가중치는 가장 보편적이며 소프트웨어 대가기준에서도 채택하고 있는 IFPUG- CPM[7]의 기준을 적용한다.

또한 보정계수 적용에 있어 IFPUG-CPM은 조정인자(VAF)로 일반적인 시스템 특성(GSC) 14가지, UCP는 기술적인 요인(TCF) 13가지와 환경적 요인(EF) 8가지가 있으나, VAF는 기능점수 모형의 설명력 향상에 크게 도움을 주지 않으며 TCP와 EF는 보정 전에 비해 ±40%의 편차를 유발시킬 수 있어 효용성이 없는 것으로 연구되고 있어 본 논문에서는 규모, 애플리케이션, 개발언어, 그리고 품질/특성에 의한 보정 등 구체적으로 세분화한 소프트웨어 사업대가 기준을 적용한다.

4.1.1 기능크기 계산

객체지향 기능크기의 측정은 (그림 6)과 같이 서비스와 클래스/객체 기능 식별을 통해 트랜잭션과 데이터 기능을 도출한다. 보정되지 않은 미조정 기능점수(UFP)의 애플리케이션 전체 크기는 서비스 기능에 가중치를 계산하고 클래스 기능에 가중치를 계산한 합으로 이루어진다.

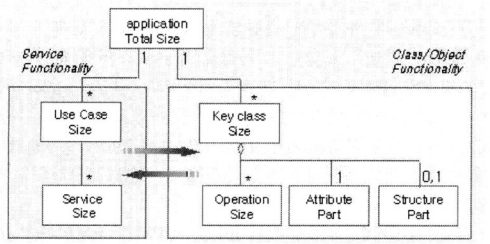
트랜잭션의 크기인 서비스 기능은 분석 활동의 산출물인 유스케이스 명세서를 통해 처리해야 할 사건흐름(기본흐



(그림 5) 기능점수 측정 구성

름, 대안흐름, 예외흐름), 사전/사후 조건, 시나리오 등의 요구사항으로 식별할 수 있으나 작성자의 능력과 정제수준에 따라 차이가 있다. 따라서 키 클래스의 오퍼레이션을 이용하여 트랜잭션 기능(EI, EO, EQ)을 식별하고 클래스의 속성 개수를 이용하여 DET를 식별하며, 클래스 간의 연관관계를 이용하여 FTR을 식별하는 것이 보다 합리적이고 객관적인 방안이라 본다.

데이터의 크기인 클래스/객체 기능은 유스케이스의 액터 중 사용자 액터와 관련된 클래스에서 ILF 개수, 시스템 액터와 관련된 클래스에서 EIF 개수를 식별하고, 클래스 구조의 관계, 즉 일반화, 집합연관, 복합연관으로 연관된 클래스를 서브그룹으로 묶고 RET 개수를 도출한다.



(그림 6) 객체지향 애플리케이션의 기능식별 구조

4.1.2 복잡도/가중치 부여

객체지향 규모의 복잡도에 적절한 가중치를 부여하기 위해 유스케이스와 클래스 기능의 복잡도/가중치는 기존 구조적인 FPA의 트랜잭션 기능과 데이터 기능의 복잡도/가중치를 적용한다. ILF, EIF와 EI, EO, EQ에 대한 복잡도/가중치는 개발 초기단계에서 규모를 측정하기 위해 소프트웨어 대가 기준에서 제시하는 평균 복잡도/가중치를 적용하고, 개발 분석단계에서 규모를 측정하기 위해 IFPUG -CMP의 정상 복잡도/가중치를 적용하여 계산한다.

4.1.2.1 평균 복잡도/가중치

평균 복잡도/가중치란 과거 수행된 애플리케이션의 기능점수 측정결과를 통계 분석하여 ILF, EIF, EI, EO, EQ에 적용된 복잡도에 대해 계산한 가중치 평균값을 말한다. 평균 복잡도/가중치는 <표 1>과 같으며, 적용은 프로젝트 제안단계 또는 프로젝트 수행초기에 기능점수 측정에 필요한 자료가 충분하지 않은 경우에 적절하게 사용한다[13].

〈표 1〉 평균 복잡도 가중치

유형	내부 논리파일	외부연계 파일	외부 입력	외부 출력	외부 조회
가중치	7.3	5.4	4.0	5.1	3.8

4.1.2.2 정상 복잡도/가중치

ILF과 EIF의 복잡도(낮음, 보통, 높음)와 기능점수 가중치는 RET의 개수와 DET의 개수에 따라 결정된다. ILF와 EIF의 복잡도와 가중치는 IFPUG-CPM의 수치를 적용한다. EI, EO, EQ 각각의 복잡도(낮음, 보통, 높음)와 기능점수 가중치는 FTR의 개수와 DET의 개수에 따라 결정된다. EI, EO, EQ의 복잡도와 가중치는 IFPUG -CPM의 수치를 적용한다[14].

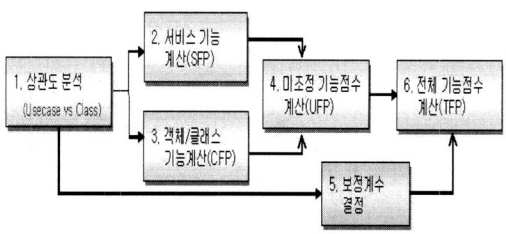
4.1.3 보정계수 적용

소프트웨어 개발 프로젝트의 생산성은 사람, 기술, 환경, 제품 등의 여러 가지 복잡성 요소에 의해 영향을 받게 되므로, 소프트웨어 기능점수만으로는 사업의 정확한 규모 측정이 어렵다. 따라서 이러한 복잡성 요소들을 감안해 정통부 소프트웨어 대가기준에서 제시하는 4가지의 보정계수를 곱하여 규모를 측정한다[11].

먼저 소프트웨어 개발 규모가 커짐에 따라 투입인력과 커뮤니케이션이 증가하여 생산성이 떨어지기 때문에 사업 규모의 증가에 따른 규모 보정이 필요하다. 그리고 같은 규모의 애플리케이션 소프트웨어라 하더라도 그 특성별 유형에 따라 개발자의 생산성이 달라지게 되므로 이러한 애플리케이션 유형에 대한 규모 비율을 곱하여 합산한 값을 보정계수라 하며 소프트웨어 대가기준의 보정계수를 적용한다. 소프트웨어 생산성은 개발에 사용된 프로그램밍 언어에 영향을 받게 되므로 언어에 따라 소프트웨어 대가기준의 보정계수를 적용한다. 마지막으로 사용자가 애플리케이션 시스템에 요구하는 특정 품질 및 특성은 소프트웨어 개발의 생산성에 영향을 미치게 되므로 사용자가 애플리케이션 시스템에 요구하는 품질 특성에 따른 소프트웨어 대가기준의 보정계수를 적용한다.

4.2 객체지향 기능점수 측정 프로세스 설계

기능점수 측정 프로세스는 유스케이스와 클래스 분석 결과를 이용하여 상관도 분석을 통해 서비스 기능 및 객체/클래스 기능을 식별하고 서비스기능과 객체/클래스 기능에 대한 제안된 복잡도/가중치를 부여하여 기능점수를 계산한다. 다음으로 두 종류의 기능점수를 합해서 미조정 기능점수를 계산하고 끝으로 제안된 보정계수를 부여하여 전체 기능점수를 계산한다. 이와 같이 제안한 기능점수 측정 프로세스가 (그림 7)에 도시되어 있다.



(그림 7) 제안 기능점수 측정 프로세스

IFPUG-CPM과 제안 프로세스의 차이점으로는 IFPUG-CPM은 구조적인 개발 절차에 기반을 두고 있으나 제안 프로세스는 최근 개발 패러다임에 부합하는 객체지향 개발 절차를 중시하고 있다. 또한 전자는 고도의 전문가적인 식견에 의한 기능 식별이 용이하도록 되어 있으나 후자의 경우는 UML이나 CBD 산출물을 이용한 상관도 분석 템플릿을 이용하여 누구나 객관적으로 식별할 수 있도록 했다. 끝으로 전자는 1979년 Albrecht의 기능점수 측정규칙을 고수하고 있는 반면 후자는 OO 개념에 부합하는 측정 규칙의 재정립을 들 수 있다.

이러한 근본적인 차이점을 기반으로 본 절에서는 기능점수 측정절차에 대해 제안한 기법들을 설명하고 수행활동 작업에 대해 기술한다.

4.2.1 1단계: 상관도 분석

서비스 기능과 객체/클래스 기능을 식별하기 위해 사전에 UML에 의해 분석된 유스케이스와 클래스 다이어그램 산출물이 필요하며 이를 기반으로 상관도 분석을 수행한다.

서비스 기능은 트랜잭션 기능으로서 IFPUG-CPM에서는 애플리케이션이 데이터를 처리하여 사용자에게 제공하는 것으로 정의하며, 사용자 도메인의 유스케이스 분석을 통하여 사용자 관점에서 시스템 도메인을 분석하는데 사용한다. 보다 정확한 애플리케이션 경계 및 서비스 기능 식별을 위해서는 클래스와 상관분석이 요구되며 시퀀스 다이어그램 및 요구사항명세서, 유스케이스명세 등의 산출물 분석이 필요하다. 본 논문의 유스케이스 대 클래스 상관도 분석을 위한 규칙은 다음과 같다.

• 작업 1: 유스케이스 모델링을 한다.

객체/클래스는 데이터 기능으로서 IFPUG-CPM에서는 사용자에게 내·외부 데이터의 요구사항을 충족시키기 위해 제공되는 기능이라고 정의하며, 사용자 도메인의 비즈니스 객체 분석을 통하여 사용자 관점에서 시스템 도메인을 분석하는데 사용한다. 클래스의 여러 유형 중 기능점수의 파일 개념에 상응하는 엔티티 클래스와 제어 클래스로 한정하며, 기능점수의 측정시점에 따라, 즉 사업초기의 요구사항 도출시 개념적 객체 모델과 요구분석 단계에서 클래스 모델을 통해 기능을 식별한다. 본 논문의 유스케이스 대 클래스 상관도 분석을 위한 작업은 다음과 같다.

• 작업 2: 객체 모형 및 클래스 모델링을 한다.

다음은 유스케이스 대 클래스 상관도 분석 작업이다. 이를 위해서는 먼저 상관도와 관련된 규칙을 정해야 하는데 이에는 객체/클래스 기능과 서비스 기능을 정하는 규칙이 있고 DET, RET, FTR과 관련된 규칙이 있다.

객체/클래스 기능은 데이터 기능으로서 논리적으로 연관된 데이터 그룹 또는 제어 정보로 ILF, EIF의 2가지 유형으로 구분되며 유스케이스의 사용자와 액터에 의해 실현되는 애플리케이션 경계 내부에서 유지되는 클래스 그룹을 ILF로 결정하고 시스템 액터에 의해 유지되는 클래스 그룹을 EIF로 결정한다. 서비스 기능은 트랜잭션 기능으로서 애플리케이션이 데이터를 처리하여 사용자에게 제공하는 EI, EO, EQ의 3가지 유형이 있으며 각 기능의 값은 유스케이스 기반으로 작성된 클래스를 분석하여 오퍼레이션 유형을 식별하여 EI, EO, EQ의 유형을 정한다.

DET는 사용자가 식별 가능한 비반복적인 유일한 필드를 의미하고 클래스의 속성이 가지는 개념이라 할 수 있으며, RET는 ILF나 EIF 안에서 사용자가 식별 가능한 데이터 요소의 서브그룹으로 정의되고 식별된 후보 객체들과 연관된 클래스에 대응된다. Fetcke[3]은 클래스 간의 관계를 집합과 복합 관계인 포함과 상속관계로 규정하였다. 따라서 DET는 ILF 또는 EIF에서 유지되는 유일한 클래스의 속성을 하나의 DET로 측정하고 ILF나 EIF의 포함관계에 있는 엔티티 클래스 그룹의 각각의 클래스 계층을 RET로 하고 포함관계가 없으면 하나의 RET로 본다. 또한 상속관계에 있는 클래스는 하나의 RET로 계산하고 부모 클래스의 속성을 자식 클래스의 DET로 계산한다. 마지막으로 트랜잭션에 의해 유지되거나 읽혀지는 파일을 FTR이라 하며 식별된 트랜잭션 내에 포함된 모든 엔티티 클래스를 각각 하나의 FTR로 계산한다. 즉 오퍼레이션만 있고 속성이 없는 제어 클래스는 FTR 개수에서 제외한다. 이러한 규칙을 바탕으로 <표 2>와 같이 상관도를 작성한다.

〈표 2〉 유스케이스 vs 클래스 상관도

Class / Use Case	C1	C2	C3	C4	C5	C6	C7	Tran. Type	FTR Count
Attribute									
RET Count									
U1									
U2									
U3									
U4									
U5									
U6									
U7									

X축에는 C1, C2 등 식별된 각각의 클래스를 배치하고 Y축에는 U1, U2 등 식별된 각각의 유스케이스를 배치한다. 유스케이스와 클래스가 교차하는 셀(Cell)에는 ILF 또는 EIF를 표시한다. 그리고 객체/클래스(Attribute, RET, FTR) 기능과 트랜잭션(EI, EQ, EI) 기능의 개수를 구한다. 본 논문의 유스케이스 대 클래스 상관도 분석을 위한 작업은 다음과 같다.

- 작업 3: 유스케이스 대 클래스 상관도를 작성한다.
- 작업 4: 시스템의 경계를 식별한다.
- 작업 5: ILF, EIF를 식별한다.
- 작업 6: Attribut와 RET를 식별한다.
- 직업 7: FTR을 식별한다.
- 직업 8: EI, EQ, EO를 식별한다.

4.2.2 2단계: 서비스 기능점수(SFP) 계산

1단계의 상관도 분석에서 식별된 DET와 FTR의 개수를 이용하여 소프트웨어 대가기준의 평균 복잡도/가중치나 IFPUG-CPM의 정상 복잡도/가중치를 부여한다. 트랜잭션 기능(EI, EO, EQ)별로 구분하여 서비스 기능점수인 SFP(Service Function Point)를 <표 3>과 같이 계산한다.

서비스 기능의 트랜잭션(EI, EO, EQ)은 유스케이스의 여러 가지 트랜잭션으로 구성되어 있기 때문에 관련된 엔티티 클래스들을 이용하거나 또는 CBD 산출물인 유스케이스 명세서의 사건흐름, GUI인 사용자 인터페이스 프로토타이핑의 구성항목 등을 이용하여 구한다.

- 작업 9: 유스케이스를 분류하여 복잡도를 계산한다.
- 작업10: 가중치를 부여하여 SFP를 구한다.

〈표 3〉 서비스 기능점수 표

구분	TYPE			Complexity			SFP
Usecase Name	EI (DET,FTR)	EO (DET,FTR)	EQ (DET,FTR)	Low	Avg	High	
U1							
U2							
U3							
U4							
U5							
U6							
U7							

4.2.3 3단계: 객체/클래스 기능점수(CFP) 계산

논리적인 데이터 그룹 기능(ILF, EIF)별로 구분하여 객체/클래스 기능점수 CFP(Class Function Point)를 <표 4>와 같이 계산한다. 기능점수 측정을 위한 클래스는 데이터 갱신(입력/수정/삭제)을 위한 엔티티 클래스와 조회나 현황을 위한 제어 클래스로 구성되는데 제어 클래스는 DET 계산에 의미가 없다.

〈표 4〉 객체/클래스 기능점수표

Class Name	TYPE		Complexity			CFP
	ILF (DET, RET)	EIF (DET, RET)	Low	Avg	High	
C1						
C2						
C3						
C4						
C5						
C6						
C7						

- 작업11: 클래스를 분류하여 복잡도를 계산한다.
- 작업12: 가중치를 부여하여 CFP를 구한다.

4.2.4 4단계: 미조정 기능점수(UFP) 계산

IFPUG-CPM에서 데이터 기능점수와 트랜잭션 기능점수를 합해서 미조정 기능점수(UFP)를 구하듯 본 논문에서도 서비스 기능점수(SFP)와 객체/클래스 기능점수(CFP)를 합해서 구한다.

- 작업13: 미보정된 서비스와 클래스 기능점수를 합한다.

4.2.5 5단계: 전체 기능점수(TFP) 계산

전통적인 FPA에서는 측정된 미조정 기능점수(UFP)와 조정인자(VAF)를 통해서 전체 기능점수(TFP)를 예측하고 있으나 본 논문에서는 조정인자 대신에 소프트웨어 대가기준의 보정계수(규모 보정, 애플리케이션 유형 보정, 언어 보정, 품질/특성 보정)를 곱하여 전체 기능점수를 계산한다. 보정계수는 발주단계에서는 제안요청서(RFP)를 통해 판단할 수 있고, 분석단계에서는 요구사항명세서, 소프트웨어 아키텍처정의서를 통해 구한다.

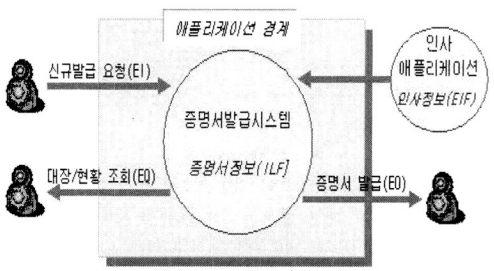
- 작업14: 보정계수를 곱하여 전체 기능점수를 계산한다.

5. 사례연구를 통한 적합성 평가

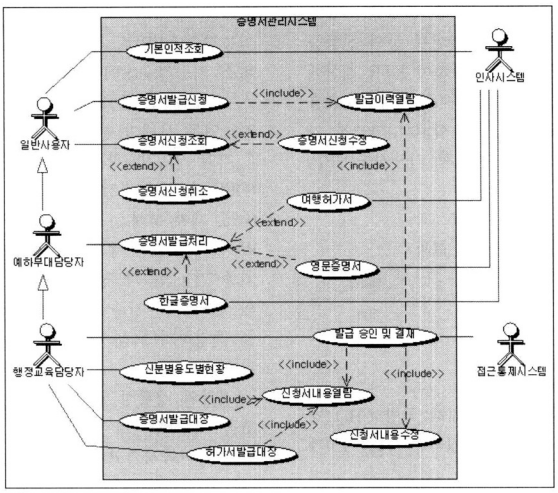
5.1 사례 연구

본 논문에서 제시하는 사례연구는 제안한 측정 프로세스 규칙에 따라 웹 체계의 '증명서발급시스템'에 대해 객체지향 기능점수를 측정하고 시스템을 개발하는 것이다.

증명서발급시스템은 국방 CBD 방법론에 따라 분석/설계 툴(Together 6.1)을 사용하여 모델링하고 산출물을 생산한다. 시스템 개발은 객체지향 언어인 HTML과 Java를 사용하며, 정보시스템 기능 구성요소는 (그림 8)과 같다. 증명서발급시스템의 증명서 종류에는 한글과 영문으로 구분된 복무확인서, 재직확인서, 전역예정증명서가 있으며 별도로 한글로 된 국외여행허가서가 있다. 발급신청은 누구나 할 수 있고 발급(출력)은 상급부대에서 승인하면 지정된 예하부대 담당자가 발급할 수 있다. 발급승인에 대한 결재는 교육담당자와 팀장까지이고, 교육담당자가 업무의 모든 통제 권한을 가지며 일반 사용자와 예하부대담당자의 업무기능을 상속받는다.



(그림 8) 정보시스템 기능 구성요소



(그림 9) 유스케이스 다이어그램

5.1.1 객체지향 분석

서비스 기능을 식별하기 위해 (그림 9)과 같이 유스케이스 모델링을 하고 유스케이스 명세서를 작성한다. 클래스 기능을 식별하기 위해 분석가능 정도 및 측정시점에 따라 개발 초기 단계에서는 개념적 객체 모델링을 하고 분석단계의 구체화 정도에 따라 상세한 클래스 모델링을 하고 클래스 명세서를 작성한다.

5.1.2 착수단계의 기능점수 계산

착수단계에서는 유스케이스와 개념객체에 대한 기능을 다음 <표 5>와 같이 식별하고 <표 1>에서 제시한 소프트웨어 대가기준의 평균 복잡도 가중치를 사용하여 기능점수를 계산한다. 단 제어 개념 객체는 계산될 속성이 없으므로 개념 객체 개수에서 제외한다. SFP와 CFP는 각각의 기능 개수에 평균 가중치를 곱해서 계산하고 UFP는 SFP의 합과 CFP의 합을 더한 점수이며 그 결과는 <표 6>과 같다.

〈표 5〉 착수단계 유스케이스와 개념객체 기능식별

유스케이스	Tran. Type
U1:기본인적조회	EQ
U2:증명서발급신청	EI
U3:증명서신청조회	EQ
U4:증명서신청수정	EI
U5:증명서신청취소	EI
U6:발급이력열람	EQ
U7:발급승인/결재	EQ
U8:신청서내용열람	EQ
U9:신청서내용수정	EI
U10:증명서발급처리	EO
U11:한글증명서	EO
U12:영문증명서	EO
U13:여행허가서	EO
U14:신분별용도현황	EO
U15:증명서발급대장	EO

개념 객체	Data Type
C1:인사정보	EIF
C2:증명서신청	ILF
C3:증명서세부	ILF
C4:허가서세부	ILF
C5:승인결재	ILF
C6:증명서	X
C7:대장현황	X
C8:서명정보	EIF

〈표 6〉 착수단계 기능점수 측정 결과

기능점수	SFP	CFP	UFP
	70.7	40.0	110.7

5.1.3 분석단계의 기능점수 계산

분석단계에서는 <표 7>과 같이 유스케이스 대 클래스 상관도를 이용하여 상관도 분석을 실시한다.

다음으로 <표 3>의 서비스 기능점수표를 이용하여 SFP를

〈표 7〉 유스케이스 vs 클래스 상관도 계산

클래스 / 유스케이스	C1 인사 정보	C2 증명 신청	C3 증명 세부	C4 허가 세부	C5 승인 결재	C6 증명 서	C7 대장 현황	C8 서명 정보	Tran. Type	FTR Count
Attribute Count	11	19	5	10	10	0	0	2		
RET Count	1	2	2	2	1	1	1	1		
U1:기본인적조회	EIF								EQ	1
U2:증명서발급신청		ILF	ILF	ILF					EI	3
U3:증명서신청조회		ILF							EQ	1
U4:증명서신청수정		ILF	ILF	ILF					EI	3
U5:증명서신청취소		ILF	ILF	ILF					EI	3
U6:발급이력열람		ILF			ILF				EQ	2
U7:발급승인/결재		ILF			ILF			EIF	EQ	3
U8:신청서내용열람		ILF	ILF	ILF					EQ	3
U9:신청서내용수정		ILF	ILF	ILF					EI	3
U10:증명서발급처리		ILF			ILF				EO	2
U11:한글증명서	EIF	ILF	ILF		ILF	ILF×			EO	4
U12:영문증명서	EIF	ILF	ILF		ILF	ILF×			EO	4
U13:여행허가서	EIF	ILF		ILF	ILF	ILF×			EO	4
U14:신분별용도현황		ILF			ILF		ILF×		EO	2
U15:증명서발급대장		ILF			ILF		ILF×		EO	2
U16:허가서발급대장		ILF			ILF		ILF×		EO	2

계산한다. 이때 트랜잭션의 DET를 식별하기 위해서는 유스케이스 명세서, 사용자 인터페이스 프로토타이핑 등의 산출물이 필요하다. 단 ILF×는 제어클래스에 해당하여 FTR 계산에서 제외한다. 그런 다음 <표 4>의 객체/클래스 기능점수표를 이용하여 CFP를 계산한 후 SFP와 CFP를 합해서 미조정 기능점수(UFP)를 계산한다. 분석단계의 미조정 기능점수(UFP)는 <표 8>과 같다.

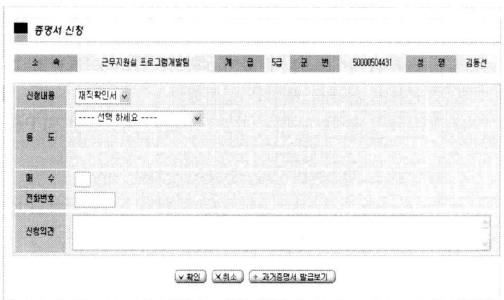
〈표 8〉 분석단계 기능점수 측정 결과

기능점수	SFP	CFP	UFP
	67	38	105

5.1.4 인도단계의 기능점수 계산

사례연구 대상인 '증명서발급시스템'에 대해 국방 CBD 방법론에 의해 분석·설계하여 실제 구현한 결과는 (그림 10)과 같다.

이를 기반으로 제안한 방법의 결과와 예측기법의 정확성을 비교분석하기 위하여 IFPUG-CPM의 기능점수 기법을 적용하여 미조정 트랜잭션 기능점수(TUFP)와 미조정 데이터 기능점수(DUFP)를 구하고 전체 미조정 기능점수(UFP)를 구하며 그 결과는 <표 9>와 같다.



(그림 10) 시스템 구현 화면

〈표 9〉 인도단계의 기능점수 측정 결과

기능점수	TUFP	DUFP	UFP
	70	38	108

5.2 적합성 분석 및 평가

제안한 객체지향 기능점수(OOFP) 측정 기법의 연구에 대한 적합성 평가를 위해 사례연구 등 3개 업무에 대해 계산된 객체지향 기능점수와 실제 시스템을 구현해서 계산된 기능점수의 결과를 통계 분석하여 적합성을 평가한다.

5.2.1 적합성 분석

적합성은 착수단계(ROFP), 분석단계(AOFP), 인도단계(DFP)의 기능점수 측정시점에 따라 보정계수가 동일하게 적용되는 것으로 가정하고 보정계수를 적용하지 않은 미조정 기능점수(UFP)로 평가한다. 트랜잭션 기능점수(TUFP)는 서비스 기능점수(SFP)로, 데이터 기능점수(DUFP)는 객체/클래스 기능점수(CFP)로 표현하고 측정 시점별로 계산한 결과는 3개 업무 사례의 결과는 <표 10>과 같다.

각각 업무별 착수단계(ROFP), 분석단계(AOFP), 인도단계(DFP)의 측정시점에 따라 각각 서비스 기능점수(SFP)는 64%, 객체/클래스 기능점수(CFP)는 36% 정도로 큰 오차 없이 일관성 있는 점수분포를 유지하고 있다. 또한 인도 단계의 최종 기능점수가 정확한 결과값이라고 가정했을 때 인도단계의 결과값은 기준 평균값이 되며, ROFP와 AOFP의 정확도를 측정하기 위해 사례연구를 분석한 평균편차는 <표 11>과 같다.

결과의 정확성 검증을 위해 평균편차에 대한 오차 분석을 했을 때 착수단계 및 분석단계 모두에서 3% 범위 내에서 정확도를 만족시키고 있다.

〈표 10〉 측정시점별 기능점수 측정결과 비교

업무명	단계	기능점수		
		SFP	CFP	UFP
증명서발급 시스템	ROFP	70.7	40.0	110.7
	AOFP	67	38	105
	DFP	70	38	108
인사정보 시스템	ROFP	275.8	170.0	445.8
	AOFP	278	162	440
	DFP	285	159	444
디지털진급심사시스템	ROFP	131.4	71.0	202.4
	AOFP	134	72	206
	DFP	137	74	211

〈표 11〉 측정 시점별 기능점수 결과

기능점수 / 측정시점	구분	SFP	CFP	UFP	신뢰구간 (95%)
착수단계 (ROFP)	평균점수	159.3	93.7	253	0.006
	평균편차	4.7	3.4	1.3	
분석단계 (AOFP)	평균점수	160	90.7	250.7	0.014
	평균편차	4	0.4	3.6	
인도단계 (DFP)	점수	164	90.3	254.3	

5.2.2 제안기법 평가

5.2.2.1 소프트웨어 대가기준 대비 제안 기법의 우수성

정통부에서는 국내 대표적인 기능점수 측정으로 국제표준인 IFPUG-CPM을 국내 표준으로 채택하여 소프트웨어 대가기준을 제시하고 있는데, 객체지향 정보시스템 개발에 있어 ISO/IEC 14143에서 지적하는 측정 한계 및 개발 패러다임의 변화를 기준으로 제안 기법의 상대적인 우수성을 비교 정리하면 다음 <표 12>와 같다.

〈표 12〉 제안 기법의 우수성 비교

비교기준		소프트웨어 대가기준	제안 기법
기능 규모 측정	개발 프로세스 초기부터 적용	프로젝트 초기에 기능 식별 곤란	유스케이스와 객체 모델을 통해 초기부터 측정 가능
	생명주기 전체에 일정하게 적용	프로젝트 진행의 단계별로 측정 기법 미제시	초기단계 및 분석단계에서 측정기법 제시
	사용자에 의한 충분한 이해	데이터 기능과 트랜잭션 기능 식별 어려움	UML 표기법에 의한 사용자 이해 쉽고 기능 식별 용이
기능점수 식별의 객관성		측정하는 개인 능력에 따라 많은 오차 발생	표준화된 표기법에 의한 객관적 기능 식별
개발 패러다임 변화에 대응		객체지향 정보시스템 개발에 직접적용 곤란	유스케이스와 클래스의 상관도 분석으로 직접 적용

5.2.2.2 객체지향 기법 대비 제안 기법의 우수성

선행 연구된 객체지향 기법들은 전통적인 기능점수 측정 기법의 전형적인 틀에서 벗어나 객체지향 개념의 새로운 측정 기법을 제시하고 있는데 이러한 기법들에 비해 상대적인 우수성은 다음과 같다.

첫째, UCP 측정 기법은 유스케이스를 통해 트랜잭션 기능인 동적인 측면만 강조하고 있어 정확한 측정이 어렵고 측정결과에 있어 기능점수의 오차가 크게 발생하고 있으며 시퀀스 모델을 이용한 기능점수 측정 기법에서는 유스케이스에 대한 논의가 전혀 없고 데이터 기능 식별이 지나치게 단순화되어 측정결과에 대한 신빙성이 떨어지고 있는 반면, 본 제안 기법은 유스케이스를 통한 동적인 측면과 클래스를 통한 정적인 측면에서 기능점수를 측정토록 하여 측정의 정확성과 신뢰성을 높였다.

둘째, OO-Jacobson 접근을 통한 FPA에 사상 기법은 객체지향 개념의 고려가 거의 없이 단순히 기능점수 측정에 있어 IFPUG-CPM의 규칙에 비례적인 할당을 하도록 제시되어 있는 반면, 본 제안 기법은 객체지향 접근 원리를 이용하여 유스케이스와 클래스의 상관도 분석을 통해 트랜잭션 기능과 데이터 기능을 식별 할 수 있도록 하여 더욱 객체지향적인 개념에 맞도록 기능점수를 측정 할 수 있도록 했다.

셋째, 다른 객체지향 기법에 비해 본 제안 기법은 실무 적용이 가능토록 생산된 요구분석 산출물을 그대로 활용하여 적용할 수 있도록 상관도 분석 템플릿을 제공하여 누구나 쉽게 분석결과에 대한 수치만 대입하면 기능점수 측정을 가능토록 했다.

6. 결 론

본 논문에서는 객체지향 정보시스템에 대한 효율적인 기능점수 측정 기법을 제안하였다. 이를 위해 우선 FPA를 OO 개념모델에 적용하기 위해 FPA 개념모델과 OO-FPA 개념모델을 비교하여 서비스 기능이 트랜잭션 기능과 객체/클래스 기능이 데이터 기능과 사상되는 관계를 분석하였고 이러한 접근원리를 이용하여 UML 모델링을 기반으로 한 유스케이스와 클래스의 상관도 분석을 통해 서비스 기능과 객체/클래스 기능을 식별하고 트랜잭션 기능점수와 데이터 기능점수를 구하여 객체지향 기능점수를 측정하는 기법을 제시하고 적합성을 평가하였다.

평가 결과 본 논문에서 제안한 기능점수 측정 방법은 프로젝트 초기단계 시점(ROFP)과 분석단계 시점(AOFP)에서 기능점수를 측정할 수 있도록 하였고 유스케이스 대 클래스 상관도를 이용하는 측정방법을 제안하였기 때문에 UML을 이해하는 누구든지 쉽게 기능점수 측정을 할 수 있도록 하였다.

참 고 문 헌

[1] 배문식, 「소프트웨어 사업대가 기준」, 한국 소프트웨어 진흥원, 2006.

[2] J. Kammelar, "A Sizing Approach for OO- Environments," pp.1-7, 2000.

[3] T. Fetcke, A. Abran and Tho-Hau Nguyen, "Mapping the OO-Jacobson approach into function point analysis," pp.2-6, August, 1997.

[4] Gustav Karner, "Resource Estimation for Objectory Projects," pp.3-7, September, 1993.

[5] 한국정보통신기술협회, 「기능 사이즈 측정 정의 표준: ISO/IEC 14143-1」, 한국정보통신기술협회, 1999.

[6] 컴포넌트비전, 「실전 CBD Project」, 영진닷컴, 2005.

[7] David Garmus and David Herron, *Function Point Analysis: Measurement Practices for Successful Software Projects*, Addison-Wesley, 2001.

[8] T. Uemura, S. Jusumoto and K. Inone, "Function Point Measurement Tool for UML Design Secification," 5th International Software Metric Symposium, pp.62-69, 1999.

[9] Donald J., "Web Object Counting Conventions," Reifer Consultants Inc., pp.1-3, March, 2002.

[10] 김홍식, 「공공정보화 규모측정 개선 방안」, 한국정보기술원가표준원, 2007.

[11] 이충희, 「소프트웨어 사업대가 기준과 기능점수」, 한국소프트웨어진흥원, 2006.

[12] Dekkers T., Kammelar, J., "A Function Sizing Meta Model," pp.1-3, 2002.

[13] 한국소프트웨어진흥원 「소프트웨어 사업대가의 기준 정부통신부 고시 제2006 18호」, 정보통신부, 2006.

[14] IFPUG, *Function Point Count Practice Manual, Release 4.2*, International Function Point Users Grup, 2004.

김 동 선

e-mail : reo114@naver.com
1987년 한양대학교 산업공학(공학사)
1995년~현 재 국군기무사령부 재직
2006년~현 재 국방대학교 정보관리학과 석사과정
관심분야 : 소프트웨어 공학, IT 프로젝트 관리, 개발방법론

윤 희 병

e-mail : hbyoon@kndu.ac.kr
1983년 해군사관학교(이학사)
1986년 연세대학교(공학사)
1991년 미국 해군대학원 전산공학(석사)
1998년 미국 Georgia Institute of Technology 전산공학(박사)
2002년~현 재 국방대학교 전산정보학과 부교수
관심분야 : 임베디드 소프트웨어, 소프트웨어 공학, 소프트웨어 테스팅, NCES, SOSCOE, NOA