요구사항으로부터 기능점수를 측정하기 위한 체계적인 방법

양원석*, 정창해**, 최순황**, 황만수***, 박수용**

A Systematical Method For Counting Function Point From Requirements

WonSeok Yang*, ChangHae Jung**, SoonHwang Choi**, ManSoo Hwang*** and SooYong Park**

ABSTRACT

Our research proposes how to, systematically, count function point from initial functional requirements based on natural language. Gradually, Function Point Analysis is used to overcome the limitation of LOC (Line Of Code) for estimating software size. Moreover, it plays an important role in cost management. Function point is derived from initial requirements and is determined by experts who have an education for function point. However, currently there are few researches to count function point by systematic or automatic rules.

Through extending our proposed method, we expect that function point is able to be counted automatically or semi-automatically. This would be our future research

Key Words : 기능점수, 요구사항, 데이터 기능, 트랜잭션 기능, 규칙, 프로세스

1. 서론

소프트웨어는 얼마만큼의 가치를 창출하는 가에 의해서 개발여부가 결정된다. 하지만 개발이 결정되었다고 하더라도 소프트웨어의 특성상 완전히 개발되기 전까지는 그 진행과정을 알기가 어렵다. 즉 얼마만큼의 비용과 시간, 그리고 노동이 필요한지를 가늠하기가 어렵게 된다. 소프트웨어 개발비용 산정은 소프트웨어 시스템을 구축하는데 소요되는 시간과 비용 및 노력을 예측하는 과정이다[1].

20년 전부터 소프트웨어 비용 산정은 중요한 관심의 대상이 되어 왔으며 집중적인 연구 활동에도 불구하고 아직까지도 일반화된 결론을 유도하지 못하고 있는 실정이다[2,3].

비용 산정을 위해서 사용되는 방법은 크게 두 가지, 즉 본수 [5,6] 방식과 파라미터 방식인 기능점수 방식으로 나누어진다. 여기서 본 수 방식은 많은 문제점을 가지게 되어서 현재는 기능점수 방식이 많이 사용되고 있다[4,7]. 이 기능점수 방식은 사용자의 관점에서 비용을 산정하는 방식으로 데이터 관리 위주인 경영정보 시스템 소프트웨어에 적용이 쉽고 또한 구조적 개발 방법에 적합한 기능성에 기반을 두고 있어서 트랜잭션 처리나 파일 형태에 적합하다.

기능점수는 사용자 요구에 근거한 정확한 규모를 산정함으로써 제안 및 계약의 합리성을 추구할 수 있고 사업관리의 투명성을 가질 수 있다. 또한 일관된 기준에 의해서 비용이 산정됨으로 대외 경쟁력 제고의 기회를 마련한다[9]. 그 외에도 객관적인 비용 산정을 할 수 있다는 점에서 기능점수는 중요하다. 하지만, 기능점수로 소프트웨어 비용을 산정하기 위해서는 기능점수 전문가에 의해서 비용이 산정되고, 또한 요구사항과 기능점수 사이의 관계에 대한 연구도 미흡하고 특히 자동적으로 또는 체계적으로 요구사항에서 기능점수를 측정하는 것에 관한 연구도 미미하다. 따라서, 본 논문은 요구사항으로부터 직접적으로 기능점수를 추출하기 위한 정형화된 표현 규칙과 프로세스를 제안한다

2. 관련 연구

기존의 본 수 및 LOC(Line Of Code) 방식과는 달리 기능점수 (Function Point)는 소프트웨어를 개발하는데 필요한 기능적인 요소를 측정하는 것이다. 따라서 소프트웨어의 코딩을 위한 언어나 기술들은 기능적인 요소를 측정할 시에는 배제된다. 그림 1에서처럼 IFPUG-CPM에서 제공하는 standard 버전에서는 7 단계로 구성 되어진다[7] : 측정 유형 결정, 측정 범위와 어플리케이션 경계 식별, 데이터 기능 측정, 트랜잭션 기능 측정, 미조정 기능점수 결정, 조정인자 결정, 조정 기능점수 측정.

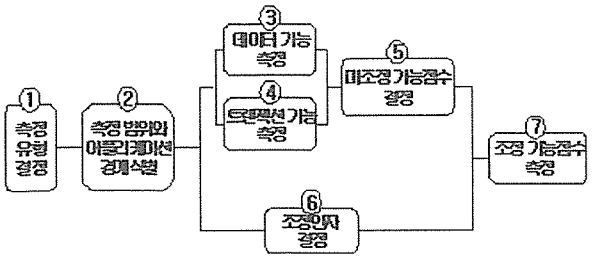

그림 1. 기능점수 측정 단계

단계 1. 측정 유형 결정

기능점수 측정은 프로젝트나 어플리케이션에 관련이 있고, 여기에는 다음 3가지 유형이 있다: 개발 프로젝트 기능점수 측정, 개선 프로젝트 기능점수 측정, 어플리케이션 기능점수 측정. 즉, 시스템을 개발하는데 있어 처음부터 개발을 하는지 추가적으로 작업을 하게되는지, 다른 어플리케이션을 사용하여 작업을 하는지를 구분하는 작업이다.

단계 2. 측정 범위와 어플리케이션 경계 식별

측정 범위란 특정한 기능점수 측정에 포함될 기능을 정의한다. 어플리케이션 경계란 측정 대상 소프트웨어와 사용자간의 경계선을 나타낸다. 인사관리 시스템을 예로 들면 인사관리시스템과 외부의 통화 어플리케이션과의 경계, 그리고 인사관리시스템과 사용자간의 경계등을 말할 수 있다.

단계 3. 데이터 기능 측정

데이터 기능은 사용자의 내부 및 외부 데이터 요구를 충족시키기 위하여 제공되는 기능을 의미하는 것으로, Internal Logic File (ILF)와 External Internal File (EIF)로 구분된다.

ILF는 어플리케이션 경계 안에서 유지되어지는논리적 관련 데이터 그룹이나 제어 정보로, 사용자가 식별 가능한 것이라야 한다. 내부 논리파일의 주요 의도는 측정 대상 어플리케이션의 하나 또는 그 이상의 단위 프로세스를 통하여 유지되어지는 정보를 갖는 것이다.

EIF는 측정 대상 어플리케이션이참조하고 경계 밖의 다른 어플리케이션 안에서 유지되어지는 논리적 관련 데이터 그룹이나 제어 정보로, 사용자가 식별 가능한 것이라야 한다. 외부연계파일로 측정되는 것은 반드시 다른 어플리케이션의 내부논리파일로 존재해야 한다.

각각의 ILF와 EIF는 File Type Reference(FTR)과 Data Element Type(DET)로 구성이 되어 있다.

단계 4. 트랜잭션 기능 측정

트랜잭션 기능은 사용자의 데이터 처리요구를 충족시키기 위하여 제공되는 기능을 의미하는 것으로, External Input (EI), External Output (EO), External Inquiry (EQ)로 구분된다.

EI는 어플리케이션 경계 밖에서 들어오는 데이터나 제어 정보를 처리하는 단위 프로세스로서, 그 주요 의도는 하나 또는 그 이상의 내부논리파일을 유지하거나 시스템의 동작에 변화를 주는데 있다.

EO는 어플리케이션 경계 밖으로 데이터나 제어 정보를 내보내는 단위 프로세스로서, 그 주요 의도는 데이터나 제어 정보의 조회와는 달리 또는 이를 수행한 후에 처리 로직을 통하여 사용자에게 정보를 제공하는데 있다. 처리 로직에는 최소한 하나 이상의 수학적 공식이나 계산 또는 파생 데이터의 생성 등을 갖고 있어야 한다.

EQ는 어플리케이션 경계 밖으로 데이터나 제어 정보를 내보내는 단위 프로세스로서, 그 주요 의도는 데이터나 제어 정보 조회를 통하여 사용자에게 정보를 제공하는 데 있다. 처리 로직에는 수학적 공식이나 계산 또는 파생 데이터의 생성 등이 없어야 하고, 내부논리파일을 유지하거나 시스템의 동작을 변경하지도 않아야 한다.

단계 5. 미조정 기능점수 결정

미조정 기능점수 측정은 프로젝트 또는 어플리케이션을 통해 사용자에게 제공될 측정 가능한 특징 기능들을 반영한다. 어플리케이션의 특정한 사용자 기능이란, 어플리케이션의 '무엇'이 사용자에게서 제공되느냐로 평가되는 것이지 "어떻게" 제공되느냐가 아니기 때문에 오직 사용자가 요구하여 정의된 컴포넌트만을 측정한다.

단계 6. 조정인자 결정

조정인자 (VAF)는 어플리케이션 사용자에게 제공되는 일반적 기능을 지칭하는 것으로, 어플리케이션의 일반 기능을 평가하는 14개의 일반 시스템 특성(GSC)으로 구성되어 있다. 각각의 특성에는 어플리케이션에 대한 영향 정도를 결정하는데 도움을 주는설명이 포함되어 있다.

단계 7. 조정 기능점수 측정

조정 기능 점수는 개발 프로젝트나 개선 프로젝트또는 어플리케이션 별로 고유한 공식을 사용하여 측정된다.

본 논문은 요구사항으로부터 기능점수를 추출해 내기 위한 규칙과 프로세스를 제시한다. 위의 7가지 단계에서 기능점수를 측정하는 것에서 데이터 및 트랜잭션 기능 측정을 하는 것이 핵심 사항이다. 따라서 본 논문에는 데이터 기능에서 ILF 및 EIF를 추출할 수 있는 방안 및 트랜잭션 기능 측정에서 EI, EO, EQ를 추출해 낼 수 있는 방안을 제시할 것이다.

규칙 1	정의	모든 요구사항은은 아래의 형식에 맞추어서 정리되어야 한다. '주어 + 목적어 + 방향 + 동사'
	해설	요구사항은 시스템에 행해지는 혹은 시스템으로부터 행하여지는 작업에 관한 상세 요구사항을 의미 한다
	예제	(사용자는)주어 (입금내역을)목적어 (시스템으로부터)방향 (조회한다)동사
규칙 2	정의	'주어'는 기능을 행하는 주체 혹은 기능이 행하여 지는 주체를 의미한다
	해설	주어는 사용자, 시스템, 다른 곳에서 이미 개발된 시스템이 될 수 있다.
	예제	(사용자는)주어 현금을 시스템에 입금한다
규칙 3	정의	'목적어'는 데이터 기능에서의 ILF, EIF에 해당하는 논리적으로연관된 데이터 그룹을 의미한다
	해설	데이터에 사용되는 ILF, EIF를 명시한다
	예제	관리자는 (사용자 정보를)목적어 시스템으로부터 변경한다
규칙 4	정의	'방향'은 데이터 기능에서 ILF, EIF의 구분을 명확히 하기 위한 내용을 의미한다
	해설	사용되어지는 데이터가 내부에 있는지 외부에 있는지를 구분할 수 있는 요소가 되어진다
	예제	사용자는 현금을 (시스템으로부터 다른 시스템으로)방향 전송한다
규칙 5	정의	'동사'는 트랜잭션 기능에서의 EI, EO, EQ를 선별하는 기준이 되는 동사를 의미한다
	해설	데이터를 입력하는 것인지, 조회하는 것인지, 수학공식이나 계산을 필요로 정보를 원하는 것인지를 구분하는 동사를 의미한다
	예제	관리자는 사용자 정보를 시스템에 (등록한다)동사

Table 1 요구사항 분석에 관한 규칙

3. 접근 방법

IFPUG가 제안하고 있는 기능점수의 측정 단계는 7가지 단계를 포함한다 (그림 1).
단계 1부터 5까지의 활동은 국제적 기준에 의해서 인증되어 있으나, 나머지 단계 6, 7의 활동은 일반적으로 넓게 사용되고 있지 않다. 단계 1에서의 활동은 초기 요구사항으로부터 직접적으로 유도되고, 단계 2에서의 활동은 다양한 방법들이 각각의 환경(도메인, 어플리케이션, 제약사항 등)으로 인하여 적용될 수 있어서 표준화하기가 힘들다. 단계 5는 단계 3과 4의 활동으로 인한 결과이다. 그러므로, 우리는 단계 3과 4의 활동을 체계적으로 수행할 수 있는 방법을 제시한다.

본 논문에서 제안된 방법은 자연어를 기반으로 한 초기의 기능적 요구사항으로부터 체계적으로 기능점수를 측정하는 방법이다. 즉, IFPUG에서 정의된 5가지의 기능 타입에 따라서 초기의 요구사항이 분류되고, 이 분류된 기능적 요구사항의 관점에서 데이터와 트랜잭션 기능점수를 측정할 수 있는 프로세스 및 규칙이 정의된다. 본 논문은 요구사항을 직접적으로 기능점수에 매핑시킬 수 있는 프로세스와 규칙을 찾는데 초점을 두고 있다. 또한, 본 논문은 우리가 발견한 규칙에 의해 재정의된 요구사항을 위한 형식도 제시한다.

3.1 요구사항 분석에 관한 규칙

요구사항은 자연어로 구성이 되어진다. 기능 점수를 구하기 위해선 데이터 기능에서의 ILF, EIF를 선별할 수 있어야 하고 트랜젝션 기능에서의 EI, EO, EQ를 선별할 수 있어야 한다. 따라서 자연어로 구성되어진 요구사항에서 기능 점수를 위해 필요한 정보를 추출하려면 다음과 같은 규칙에 맞추어서 요구사항이 정리되어야 한다.(Table 1)

3.2 데이터 및 트랜잭션 기능점수를 얻기 위한 프로세스

데이터 및 트랜잭션 기능점수에 대한 모든 것은 타겟 시스템에 있는 ILF, 참조 시스템에 있는 EIF 그리고 사용자, 시스템, 다른 시스템 사이의 트랜잭션과 관련되어 있다 (그림 2).

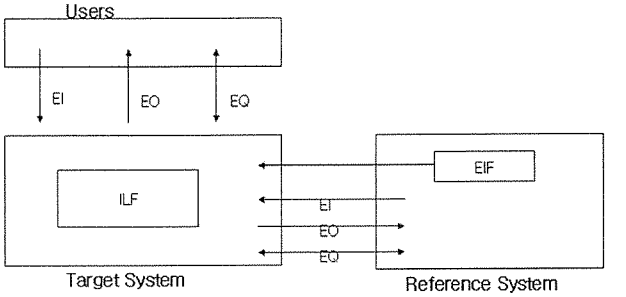

그림 2 기능점수의 트랜잭션

EI, EO 그리고 EQ 같은 트랜잭션의 기능점수는 FTR (File Type Reference)와 DET에 의해 결정되고, ILF와 EIF 같은 데이터의 기능점수는 RET와 DET에 의해 결정된다 (그림 3).

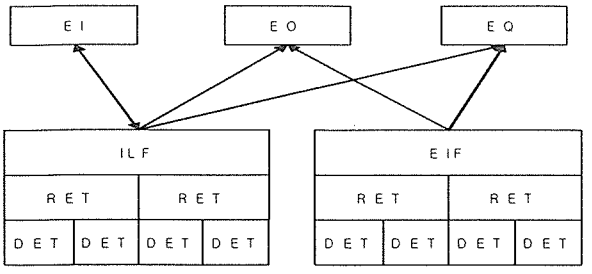

그림 3. 기능점수의 계산에 필요한 요소들

그러므로, 요구사항으로부터 데이터 및 트랜잭션 기능점수를 체계적으로 측정하기 위해서 기존의 요구사항의 재정의 및 제한이 필요하다. 제안된 규칙에 따라 재정의된 요구사항을 그림 4의 미조정 기능점수 측정 절차를 통하여, 데이터 및 트랜잭션 기능점수를 체계적으로 측정할 수 있다. 요구사항으로부터 데이터 및 트랜잭션 기능점수의 체계적 측정을 위한방법은 다음의 프로세스를 따른다 (그림 4).

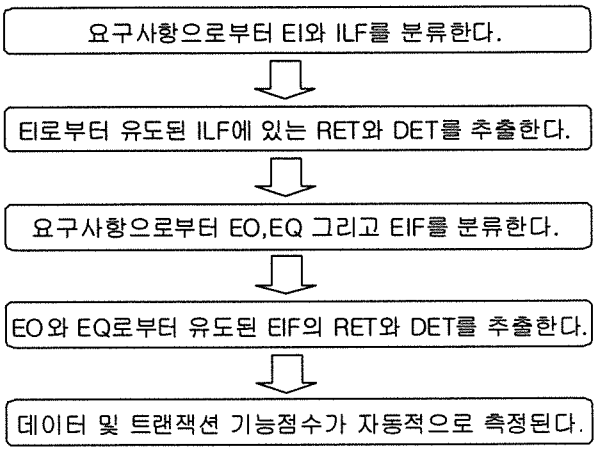

그림 4. 미조정 기능점수 측정 절차

● 스텝 1 요구사항으로부터 EI와 ILF를 분류한다.

EI 트랜잭션과 관련된 요구사항을 우선적으로 분류하고, 이에 대한 ILF를 식별한다. 하나의 EI는 오직 하나의 ILF에 대응되어서, EI 트랜잭션은 단지 하나의 FTR을 가진다.

은행관리 시스템의 예를 들면, 처음 신규 고객의 등록과 같은 경우, 다음과 같은 요구사항이 있다.

"신규 고객의 등록은 매니져(사용자)에 의해서 등록되고, 신규고객의 인적사항(이름, 주민등록번호,주소, 성별, 생년월일, 전화번호)과 신규고객에 대한 계좌정보(계좌번호, 비밀번호, 등록날짜, 예금잔액)이 입력되야 한다."

이것은 신규고객을 등록하는 것이기 때문에 EI 트랜잭션이 되고 신규고객의 정보는 하나의ILF가 된다. 이 EI 트랜잭션을 위한 재정의되고 제한된 요구사항이 제한된 형식에 따르면 다음과 같이 재정의된 요구사항이 얻어진다.

EI 트랜잭션 요구사항
(매니져는)주어 (신규 고객정보를)목적어 (타겟 시스템으로)방향 (등록한다.)동사

위의 재정의된 요구사항을 통해, 하나의 EI 트랜잭션과 하나의 ILF (고객 정보)가 얻어지게 된다. 이와 같은 스텝을 반복해서 EI 트랜잭션에 관련된 모든 요구사항을 식별해 내면 EI에 관련된 모든 ILF를 얻을 수 있다(그림 5).

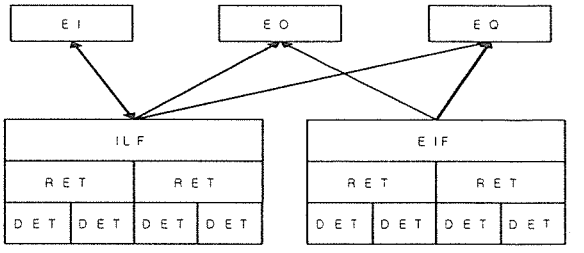

그림 5. 스텝 1의 결과물

● 스텝 2 EI 트랜잭션으로부터 유도된 ILF에 있는 RET와 DET를 추출한다.

ILF의 데이터 기능점수를 얻기 위해서는 ILF내에 있는 RET와 DET가 필요하다. RET와 DET를 얻기 위해서, 스텝 1에서 얻은 ILF를 분석해야 한다. 이과정에서 RET와 DET의 정보도 식별되어야 한다. 그러므로, 다음과 같은 형식을 통해서 ILF에 있는 RET와 DET를 식별할 수 있다. (그림 6)

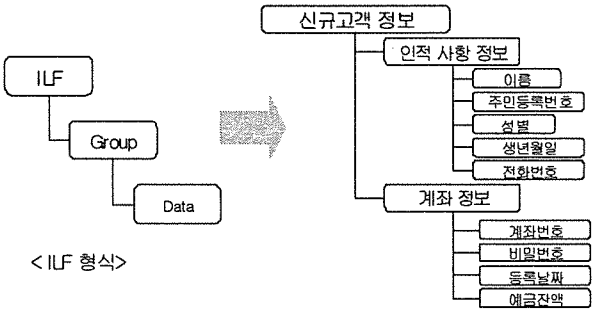

그림 6. ILF 형식의 예

스텝 1에서 보여준 신규 고객등록에 관련된 요구사항의 예를 들면, 그림 7과 같은 형식으로 나타난다. ILF의 형식은 계층을 가지는데, 이것은 ILF, 그룹 그리고 데이터로 구성된다. 그룹은 ILF에 있는 RET를 나타내고, 데이터는 ILF에 있는 DET를 역시 나타낸다. 그 결과, ILF에 대한 RET와 DET를 구할 수 있으므로, ILF의 복잡도를 구할 수 있다 (그림 7).

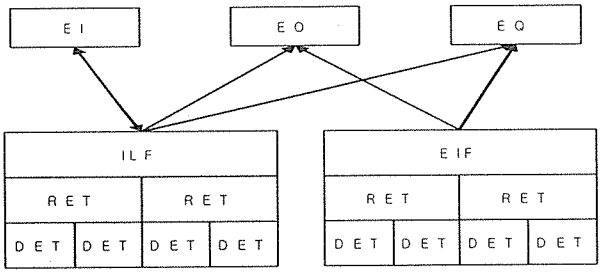

그림 7. 스텝 2의 결과물

● 스텝 3 요구사항으로부터 EO, EQ 및 EIF를 분류한다.

EO와 EQ에 관련된 요구사항을 분류한다. EO 또는 EQ 트랜잭션은 역시 단위 프로세스이며, ILF 또는EIF에 연관되어 있다. 각각의 트랜잭션은 ILF와 EIF에 관련되어 있고, 이미 스텝 1을 통해서 모든 ILF가 식별되어져 있다. 따라서, 만약 제안된 형식에 있는 '목적어'또는 '방향' 부분에 나타난 것이 스텝 1에서 나타난 것과 같은 파일이 있다면, ILF가 되고, 같은 파일이 없다면 EIF가 된다.

은행 관리 시스템의 예를 들어 보면, 고객이 계좌의 돈을 다른 계좌로 이체하는 경우, 다음의 요구사항이 있다.

"고객(사용자)은 자신의 계좌 번호와, 비밀번호를 입력한 후 계좌이체의 거래를 선택하고 이체할 계좌의 번호를 입력하면 계좌이체가 완료된다."

이와 같은 요구사항의 경우, 계좌이체 트랜잭션은 EO트랜잭션으로 분류되고 계좌 이체할 계좌번호에 해당하는 부분이 EIF에 있는 정보가 된다. EO 트랜잭션을 위한 재정의되고 제한된 요구사항이 제한된 형식에 따라서 얻어진다

EO 트랜잭션 요구사항
(고객은)주어 (돈을)목적어 (시스템으로부터 다른 시스템으로)방향 (이체한다.)동사

위의 재정의된 요구사항에서 보면, 시스템은 고객 정보를 가지고 있고, 다른 시스템은 트랜잭션을 수행하기 위해서 참고되거나 다른 시스템 내에 있는 보, 즉 EIF의 정보를 나타낸다. (그림 8). 하지만, 어떤 파일이 ILF로서 분류되면, EO 또는 EQ 트랜잭션의 복잡도를 계산하기 위해서 EO 또는 EQ 트랜잭션과 관련된 ILF에서 필요한 DET가 반드시 식별되어야 한다. ILF의 정보는 스텝 2에서 보존된다.

이와 같은 스텝을 반복하여 EO 또는 EQ에 관련된 모든 요구사항을 분류해 내면 EO와 EQ에 관련된 모든 EIF를 식별할 수 있다.

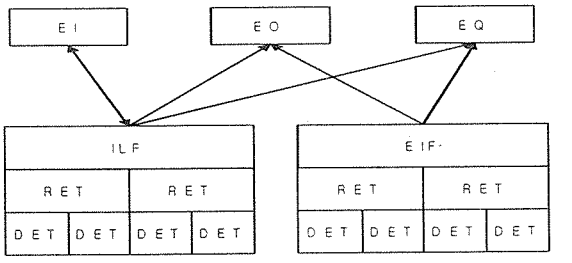

그림 8. 스텝 3의 결과물

● 스텝 4 EO와 EQ로부터 유도된 EIF에 있는 RET와 DET를 추출한다.

스텝 2와 마찬가지로, EIF에 대한 트랜잭션 기능점수를 측정하기 위해서 RET와 DET를 얻는 것이 필요하다. EIF를 위한 RET와 DET를 얻기 위해서 스텝 3에서 얻은 EIF를 분석한다. ILF와 마찬가지로, 다음과 같은 형식을 통해서 EIF에 있는 RET와 DET가 식별할 수 있다.

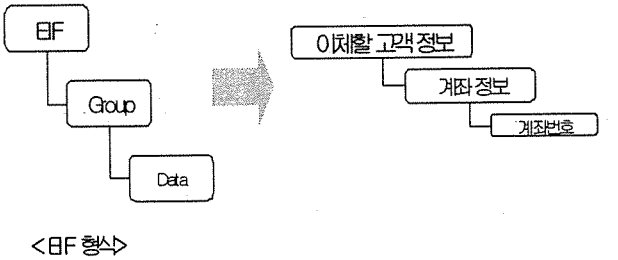

그림 9. EIF 형식의 예

스텝 3에서의 요구사항을 이용하여 예를 들면 (그림 9), 계좌이체에 대한 EO 트랜잭션은 하나의 ILF파일과 EIF 파일을 가지게 된다. ILF의 경우에는 스텝 2에서 이미 분석되었기 때문에 필요한 DET만 계산을 하면 된다. EIF인 경우에는 새로이 분석이 필요하다. EIF의 형식 역시 계층 구조를 가지게 되며, 이것은 EIF, 그룹 그리고 데이터로 구성된다. 그룹은 EIF에 있는 RET를 나타내고, 데이터는 EIF에 있는 DET를 나타낸다. 그 결과, 이 스텝을 통해서 EIF의 복잡도가 얻어지게 된다 (그림 10).

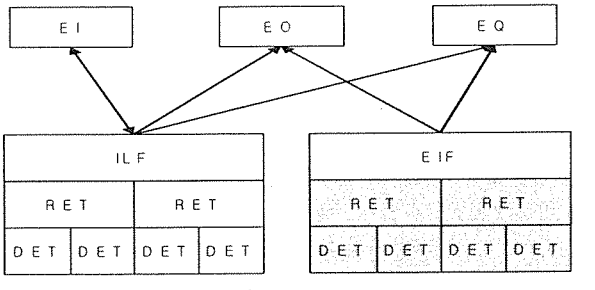

그림 10. 스텝 4의 결과물

● 스텝 5 데이터 및 트랜잭션 기능점수가 자동적으로

측정된다.

미조정 기능점수를 얻을 수 있다. 스텝 1에서 스텝 4까지를 통해 데이터 및 트랜잭션 기능점수를 얻을 수 있다. 위와 같은 스텝들을 통해서, 데이터 및 트랜잭션 기능점수의 합을 통해서 미조정 기능점수를 얻을 수 있다.

4. 결론 및 앞으로의 연구

본 논문은 요구사항으로부터 데이터 및 트랜잭션 기능점수를 측정할 수 있는 방법을 제안하였다. 자연어로 된 요구사항에서 기능점수를 추출하기 위한 방법으로 본 논문은 정형화된 표현 규칙과 프로세스를 제안하였다.

제안된 방법을 이용하면 쉽고 체계적으로 미조정 기능점수를 측정할 수 있다. 현재 사용하고 있는 기능점수의 측정 방법은 기능점수 전문가가 요구사항을 직접 분석하고 측정하는 것이다. 하지만 측정된 기능점수의 측정의 결과는 같은 조직 내에서 조차 다르게 나타날 수 있다[8]. 또한 요구사항과 기능점수 사이의 관계에 대한 연구도 미흡하고 특히 자동적으로 또는 체계적으로 요구사항에서 기능점수를 측정하는 것에 관한 연구도 미미하다. 따라서, 본 논문은 요구사항으로부터 직접적으로 기능점수를 추출하기 위한 정형화된 표현 규칙과 프로세스를 제안하였다. 이를 통해, 요구사항으로부터 기능점수를 자동적 또는 반자동적으로 측정할 수 있는 체계적인 방법에 관한기반을 제공하였다.

앞으로의 연구는 다음과 같다.

1. 제시한 표현 법칙을 표준화하기 위해서 본 논문에서 제시된 형식을 보다 구체적으로 재정의한다.
2. 실제 소프트웨어 개발 프로젝트에 적용해 본다.
3. 기능점수 전문가에 의해서 측정된 기능점수와 본 논문에서 제안한 프로세스를 통해서 측정된 기능점수를 비교 분석해 본다. 이를 통해 제안된 기능점수 측정 방법을 검사 및 평가한다.

참고문헌

[1] K. Johnson, "Software Cost Estimation: Metrics and Models", Department of Computer Science University of Calgary, Albreta, Canada, 1998.

[2] L. C. Briand. K. E. Elmam, D. Surmaann, I. Wieczork, and K. D. Maxwell, "An Assessment and Comparison of Common Software Cost Estimation Modeling Techniques", International Software Engineering Research Network, Technical Report, ISERN-98-27, 1998.

[3] L. c. Briand and I. Wieczor다, "resource Estimation in Software Engineering", International Software Engineering Research Network, Technical Report, ISERN 00-05, 2000.

[4] A. J. Albrecht: "Measuring application development productivity", Proc. Of Joint SHARE, GUIDE, and IBM Application Development Symposium (1979)

[5] Ministry of Information and Communication republic of Korea, "standard of software project cost", Notification No. 2003-14 , Ministry of Information and Communication republic of Korea (2003)

[6] Ministry of Information and Communication republic of Korea, "standard of software project cost", Notification No. 2004-8 , Ministry of Information and Communication republic of Korea (2004)

[7] IFPUG, Function Point Counting Practices Manual Release 4.1.1. Counting Practices Committee, The International Function Point Users Group(IFPUG), 2000.4.

[8] B. A. Kitchenham: "The problem with function points", IEEE Software, 14, 2 (1997)

[9] 한국 소프트웨어 산업협회. "2004년 소프트웨어사업 대가기준교육", 2004.3