

목표 및 시나리오 기반 요구사항을 이용한 기능점수 분석

(Function Point Analysis using Goal and Scenario based Requirements)

최순황[†] 김진태^{**} 박수용^{***} 한지영[†]
(Soonhwang Choi) (Jintae Kim) (Sooyong Park) (Jiyoung Han)

요약 본 논문은 목표 및 시나리오 기반의 요구사항을 이용하여 기능점수(function point)를 산정하는 방법을 제안한다. 기능점수는 소프트웨어의 규모를 계산하는 방법으로 널리 사용되고 있으며 비용계산의 기본자료가 된다. 기능점수를 산정하기 위해서는 요구사항 도출 및 분석이 선행되어야 하나 기존의 기능점수 방법론은 이를 다루지 않고 있다. 한편 시스템 개발의 초기단계에서 대부분의 요구사항은 자연어 형태로 수집된다. 목표와 시나리오 방법론은 자연어 형태의 요구사항을 사용하여 요구사항을 도출하고 분석하는 방법으로 널리 사용되고 있으며 추적성에 대한 장점을 가지고 있다. 그러므로 목표 및 시나리오 기반의 요구사항으로부터 기능점수를 산정할 수 있다면 요구사항과 기능점수 간의 추적성 관리가 쉬워진다. 이에 본 논문에서는 목표와 시나리오 기반의 요구사항으로부터 기능점수를 산정하는 방안을 제안한다. 제안된 방안은 자연어 형태로 기술된 목표 및 시나리오로부터 기능점수 분석에 필요한 규칙을 제공한다. 제안된 방안은 Order Processing System 예제를 통해 적용 방안을 설명한다.

키워드 : 기능점수, 요구사항, 목표, 시나리오, 비용

Abstract This paper proposes a method for counting function point using goal and scenario based requirements. Function Point is a software sizing method and widely used as a basis to estimate software development cost. Requirements elicitation and analysis should be performed before function point analysis but function point analysis method doesn't deal with requirements elicitation and analysis. For that reason, Function point extraction method from existing requirements method is needed and if the requirements method has advantage for traceability and elicitation, it is suitable for managing cost. Goal and scenario method is widely used as requirements elicitation and analysis. It has also good traceability. Therefore, this paper discusses a method for extracting function point from requirements text gathered using the goal and scenario based requirements elicitation technique. The proposed method aims to establish and maintain traceability between function point and requirements text. Text based function point extraction guidance rules have been developed. The proposed methodology has been applied to Order Processing System development.

Key words : Function Point, Requirements, Goal, Scenario, Cost

1. 서론

자연어 형태의 요구사항은 소프트웨어 개발 초기단계

에서 고객 요구사항을 기술하기 위해 널리 사용되고 있다 [1]. 이러한 요구사항은 소프트웨어 개발의 입력이 될 뿐만 아니라 비용 및 일정 추정의 입력도 되므로 프로젝트 관리를 위해 요구사항과 비용은 함께 관리되어야 한다[2-4]. 비용 추정 및 관리를 위해 널리 사용되는 방법 중에 기능점수분석(Function Point Analysis)이 있다. 기능점수 분석은 소프트웨어의 기능을 5가지로 구분하고 이를 점수화하여 소프트웨어 규모를 측정한다. 기능점수 분석에서 다루는 기능 기반의 소프트웨어 규모측정 방법은 다양한 비용추정 모델의 입력으로 사용되며 국제 표준

[†] 학생회원 : 서강대학교 컴퓨터학과
soonhwang@sogang.ac.kr
lunah@sogang.ac.kr

^{**} 정회원 : 삼성전자 정보통신총괄 통신연구소 선임연구원
jtae.kim@samsung.com

^{***} 정회원 : 서강대학교 컴퓨터학과 교수
sypark@sogang.ac.kr

논문접수 : 2006년 3월 22일
심사완료 : 2006년 6월 18일

및 정통부의 사업대가 기준에도 채택되어 국내외에서 널리 사용되고 있다[5-9].

기능점수 산정은 1984년 설립된 IFPUG(International Function Point User Group)에서 발간하는 표준 매뉴얼(Function Point Counting Practice Manual)[10]을 통해 표준화되어 있으며 현재 대부분의 기능점수 산정은 IFPUG에서 인증하는 자격증을 소지한 전문가들에 의해 산정되고 있다.

IFPUG의 방법론에 따라 기능점수를 산정하기 위해서는 요구사항의 도출 및 분석이 선행되어야 한다. 그러나 IFPUG의 방법론은 이에 대해 다루고 있지 않다. 그러므로 요구사항과 비용을 함께 관리하기 위해서는 기능점수분석 전문가의 도움을 받아야 한다. 예를 들어 요구사항 변경에 따른 비용영향 분석을 하기 위해서는 변경사항에 대해 기능점수 전문가에게 알리고 기능점수 재산정을 의뢰해야 한다.

따라서, 요구사항과 비용의 관리를 효율적으로 하기 위해서는 기존 요구사항 방법론으로부터 기능점수를 체계적으로 도출할 수 있는 방법이 필요하다. 기능점수 방법과 통합될 기존의 요구사항 방법론은 요구사항과 비용의 통합관리가 목적이므로 다음과 같은 특성을 가지고 있어야 한다. 첫째, 자연어 형태의 요구사항을 다루어야 한다. 비용추정은 소프트웨어 개발 초기부터 이루어진다. 그러므로 초기 단계에 수집되는 자연어 형태의 요구사항을 다루어야 한다. 둘째, 추상화 수준 개념을 제공해야 한다. 요구사항 변경에 따른 영향분석이나, 비용의 도출분석은 다양한 추상화 수준에서 분석될 필요가 있다. 예를 들어 특정 서비스를 변경하거나 서비스 내에서의 특정 기능을 변경할 때의 비용분석이 가능하기 위해서는 서비스 수준과 기능수준을 구분하는 추상화 개념이 요구사항 단계에서 정의되어야 한다. 셋째, 추적성에 장점을 지니고 있어야 한다. 비용영향분석 및 도출분석을 위해서 비용의 원인이 어떠한 요구사항으로부터 기인하는지, 또한, 그 요구사항은 어떠한 근거로부터 도출되었는지를 추적할 수 있어야 한다.

위와 같은 요건을 갖춘 기존의 요구사항 방법론은 목표 및 시나리오 방법론이 있다. 목표 및 시나리오(Goal and Scenario) 방법론은 요구사항 도출 및 분석에 널리 사용되고 있는 방법론이며 자연어 형태의 요구사항을 다루기 때문에, 고객이나 사용자와 같은 이해당사자들이 참여하는 초기 단계의 요구사항 도출 및 분석에 유리하다. 또한, 요구사항에 대한 근거를 제공하고 다양한 추상화 수준을 제공하기 때문에 각 추상화 수준에 따른 추적관리에 장점을 가지고 있다. 특정 수준의 요구사항은 상위수준의 특정 목표를 성취하기 위해 존재한다 [11-14]. 목표 및 시나리오 기반의 요구사항으로부터 기

능점수를 도출할 수 있는 체계적인 방법이 존재한다면, 다양한 추상화 수준을 가진 요구사항과 비용 간의 추적성을 관리하는데 유리하다.

본 논문의 특징은 목표 및 시나리오 기반 요구사항으로부터 기능점수를 도출할 수 있는 규칙을 제공한다는 것이다. 제안된 규칙을 통해 목표 및 시나리오 기반 요구사항과 기능점수가 연결되며 그에따라 다양한 추상화 수준을 가진 요구사항과 비용 간의 추적관리가 가능해진다.

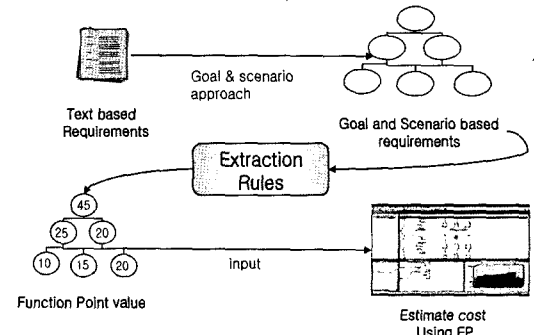


그림 1 제안된 방법의 기본 개념

제안된 방법이 요구사항과 비용을 연결하는 개념은 그림 1과 같다. 자연어 형태의 요구사항은 목표와 시나리오 기법을 통해 구조화된다. 제안된 기능점수 도출 규칙을 통해 구조화된 목표 및 시나리오로부터 기능점수를 산정한다. 산정된 기능점수를 잘 알려진 비용 모델에 입력하여 비용을 추정한다. 요구사항과 비용이 체계적인 기법과 규칙에 의해 연결되었으므로 요구사항과 비용의 연계설정을 할 수 있고, 요구사항과 비용 간의 추적성을 관리 할 수 있다.

본 논문은 다음과 같이 구성된다. 2 장에서는 기능점수와 목표 및 시나리오 분석 기법에 대한 관련연구를 다룬다. 3장에서는 본 논문에서 제안하는 목표 및 시나리오 요구사항으로부터 기능점수를 도출하기 방법을 설명한다. 4장에서는 사례연구 및 제안하는 규칙에 대해 다루고 관련 내용에 대한 토론을 한다. 5장에서 결론을 맺는다.

2. 관련연구

2.1 기능점수 분석

기능점수(Function Point)는 1979년에 개발된 소프트웨어 규모 측정방법으로 ISO-14143-1과 정통부 소프트웨어 사업대가 기준에서 표준으로 채택되어 널리 쓰이고 있다[5-9]. 또한, IFPUG(International Function Point User Group)가 1984년에 설립되어 표준 기능점

수 산정 방법인 FP CPM(Function Point Counting Practice Manual)을 발간하고 있다[10]. IFPUG에 따르면 시스템의 기능은 다음과 같이 구분된다.

- ILF(Internal Logical File): 측정 대상 시스템의 단위 프로세스를 통해 유지되는 데이터 또는 제어 정보
- EIF(External Interface File): 측정 대상 시스템의 단위 프로세스를 통해 참조되는 데이터 또는 제어정보
- EI(External Input): 하나이상의 ILF를 유지하거나 시스템의 행동을 바꾸는 단위 프로세스
- EO(External Output): 프로세스 로직을 통해 사용자에게 정보를 제공하는 단위 프로세스
- EQ(External inQuiry): ILF 또는 EIF로부터 데이터 또는 제어정보의 검색(retrieve)을 통해 사용자에게 정보를 제공하는 단위 프로세스

위에서 단위 프로세스란 사용자 관점에서 식별가능하고 의미 있는 최소 단위의 프로세스를 의미한다. 위의 기능들 중 ILF와 EIF를 데이터 기능(Data Function)이라고 하고, EI, EO, EQ를 트랜잭션 기능(Transaction Function)이라고 한다. IFPUG CPM(Counting Practice Manual)에서는 이들을 식별하고, 식별된 각 기능들에 대한 복잡도를 결정하여 기능점수를 계산하는 방법을 제공한다.

IFPUG에서 제공하는 표준을 습득한 전문가가 요구사항 문서로부터 ILF, EIF, EI, EO, EQ를 식별하고 이에 대한 복잡도를 결정하여 최종 기능점수를 계산한다. 하지만 기존 요구사항 기법으로부터의 체계적인 도출방법은 존재하지 않기 때문에 요구사항 분석과 기능분석은 분리되어 수행되어야 하고 이에 따라 요구사항 변경에 따른 기능점수 재산정과 같은 추적관리에 어려움이 따른다.

2.2 목표 및 시나리오 기반 요구사항 분석

목표 및 시나리오 기반의 요구사항 분석 방법은 요구사항 도출과 추적성에 장점을 가진 방법으로 널리 사용되고 있다[11-14]. 본 논문에서는 Kim et al이 [14]에서 제안한 방법을 사용한다. 지면이 부족한 관계로 본 절에서는 핵심 개념과 추가적인 사항만을 기술한다. 보다 자세한 내용은 [14]을 참조하길 권장한다.

목표 및 시나리오 기반 요구사항에서 요구사항의 추상화 수준은 Business Level, Service Level, Interaction Level, Internal Level 로 구분된다.

비즈니스 수준(Business Level)은 도메인에 대한 궁극적인 목표를 정의하고 기술한다. 비즈니스 레벨에서 요구사항은 비즈니스 목표로 표현된다.

서비스 수준(Service Level)에서는 도메인이 사용자에게 제공해야 하는 서비스의 범위를 정의한다. 서비스의 범위는 조직의 비즈니스 전략 또는 마케팅 전략에

의해 결정된다.

상호작용 수준(Interaction Level)에서는 서비스 범위에 대해 서비스가 실현되기 위해 사용자 또는 외부 시스템과 대상 시스템과의 상호작용을 표현한다.

내부 수준(Internal Level)에서는 상호작용을 만족시키기 위해 내부적으로 시스템이 가져야 할 기능에 중점을 둔다.

상위 수준의 목표는 시나리오에 의해 하위 수준의 목표로 분화된다. 이는 그림 2에 나타나 있다.

이때, 목표는 <Verb + Target + Direction + Way> 형식의 단문으로 기술된다. 'Verb' 와 'Target'은 반드시 필요한 성분이지만, 'Direction'과 'Way'는 때에 따라 생략 가능하다. 'Direction'은 데이터 또는 제어정보의 시작점(source) 또는 도착점(destination)을 의미한다. 'Way'는 'Mean' 또는 'Manner'를 의미한다. 'Mean'은 목표를 성취하기 위한 수단이고, 'Manner'는 목표를 성취하기 위한 방법이다. 예를 들어 '(add)_{verb} (order)_{target} (to order management application)_{direction}'에서 'order management application'은 'order'라는 객체의 도착점(destination)이다. 시나리오를 기술하는 형태는 <Subject + Verb + Target + Direction + Way> 형식을 따른다. 특별히 시나리오는 구체적으로 기술하기 위해 'Subject'를 추가하며, 목표 달성을 위해 기대되거나 가능한 행위를 작성한다. 목표와 마찬가지로 'Direction'과 'Way'는 생략 가능하다. 예를 들어 '(Order management application)_{sub} (receives)_{verb} (invoice id)_{target} (from accounting system)_{direction}'에서 'Order management application'은 'subject'이며 'invoice id' 객체의 도착점(destination)이고 'accounting system'은 'invoice id'라는 객체의 시작점(source)이다.

요구사항은 목표와 시나리오의 청크(chunk)으로 나타내며 (G,Sc) (G는 목표(Goal) Sc는 시나리오(Scenario)를 의미)로 표기한다.

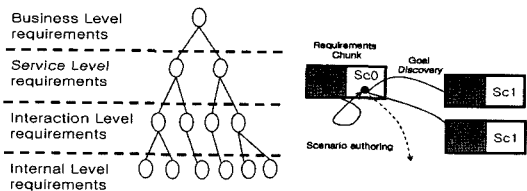


그림 2 목표 및 시나리오 요구사항 개념

3. 목표 및 시나리오 기반 요구사항으로부터의 기능점수 도출 방법

본 장에서는 목표 및 시나리오 기반의 요구사항으로

부터 기능점수를 도출하는 방법을 기술한다. 제안된 방법은 비즈니스 수준과 상호작용 수준 요구사항 청크(chunk) (G, Sc)로부터 기능점수를 도출한다. 기능점수에서의 기능은 사용자 관점에서의 데이터 기능과 트랜잭션 기능을 의미한다. 목표 지향 분석 방법에서의 이러한 관점은 상호작용 수준에서 나타난다. 상호작용 수준은 사용자 또는 외부 시스템과 대상 시스템 간의 상호작용 관점에서 목표를 기술한 것이다. 그러므로 사용자 또는 외부시스템과 대상 시스템 사이에서 전송되는 데이터(data) 정보와 동작(behavior) 정보를 포함하고 있다. 데이터 정보로부터는 기능점수에서의 데이터 기능을 도출하고 동작 정보로부터는 트랜잭션 기능을 도출할 수 있다.

우리가 제안하는 접근방법은 3단계로 이루어지며 각 단계별로 도출규칙을 제공한다. 도출 규칙은 IFPUG의 정의로부터 유도되었다. 'Step 1:Context elements 도출'에서는 기능점수 측정 범위를 설정하기 위해 사용자, 측정대상 어플리케이션, 외부 어플리케이션을 식별한다. 이는 Business goal과 <(Subject) + Verb + Direction + Way > 형태로 기술된 목표 및 시나리오의 'Subject'와 'Direction'으로부터 도출된다. 'Step 2: 데이터 기능 측정'에서는 데이터 기능인 ILF와 EIF를 식별하고 이들에 대한 복잡도를 결정한다. 이 과정은 'Target' 으로부터 식별된 데이터 정보를 그룹으로 묶어서 수행된다. 'Step 3: 트랜잭션 기능 도출'에서는 EI, EO, EQ 를 식별하고 이들의 복잡도를 결정한다. 트랜잭션 기능의 이름은 Interaction Level의 Goal에서 'Verb + Target + (direction)' 으로 정하며 세부 규칙에 종류 및 복잡도가 결정된다. 보다 자세한 설명은 세부절에 기술되어 있으며 규칙에 대한 자세한 정의 및 예제는 사례연구를 통해 설명된다. 위에서 설명한 접근방법을 도식화 하면 그림 3과 같다.

3.1 Step1: 컨텍스트 요소(Context elements) 도출

기능점수 분석을 위해서는 먼저 측정 범위와 어플리케이션의 경계를 정의하여야 한다. 경계는 측정대상 어플리케이션(Target Application)과 사용자 또는 외부 어플리케이션(External Applicatoin) 간을 구분한다.

이를 수행하기 위해서는 우선 사용자, 측정 대상 어플리케이션, 외부 어플리케이션을 식별하여야 한다. 일반적으로 측정대상 어플리케이션은 프로젝트 시작 시점에 주어지지만 프로젝트의 규모가 크고 비즈니스 기능이 분할 가능하다면 여러 개의 어플리케이션으로 분할될 수 있다. 이 경우 각 어플리케이션의 기능점수는 개별적으로 측정된다.

목표 및 시나리오 모델로부터 어플리케이션 분할을 수행하고, 측정 대상 어플리케이션, 사용자, 외부 어플리

Interaction Level

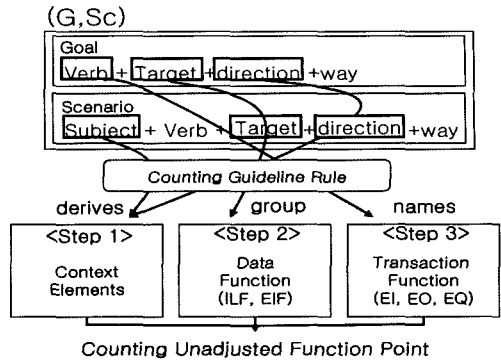


그림 3 목표 및 시나리오로부터의 기능점수 도출

케이션을 식별하기 위해 본 논문에서는 어플리케이션 분할에 대해 다루는 CA1, 에이전트 식별에 대해 다루는 CA2, 어플리케이션 분할시의 측정에 대해 다루는 CA3의 3가지 규칙을 제안하였다. 규칙에 대한 정의 및 예제는 4장 사례연구에서 기술한다.

3.2 Step2: 데이터 기능 도출

컨텍스트 분석을 통해 측정 대상 시스템의 범위를 정의 한 후에는 데이터 기능인 ILF(Internal Logical File)와 EIF(External Interface File)를 식별하여야 한다. 데이터 기능은 상호작용 수준 요구사항 청크(G, Sc)에서 'Target' 으로부터 도출될 수 있다. 'Target' 은 대상 시스템과 사용자 또는 외부시스템 사이에 상호작용하는 데이터나 제어정보를 포함한다. 이러한 데이터는 논리적으로 측정대상 어플리케이션이나 외부 어플리케이션 중 한곳에 저장된다. 이들 데이터가 측정대상 어플리케이션에서 유지되는 것이면 ILF로 외부 어플리케이션에 유지되는 것이면 EIF로 식별할 수 있다.

ILF와 EIF를 식별한 후에는 데이터의 복잡도를 계산하기 위해 RET(Record Element Type)와 DET(Data Element Type)를 식별하여야 한다. DET는 사용자가 식별할 수 있는 최소 단위의 자료요소 형태이다. DET를 사용자에게 의미 있는 그룹으로 묶으면 RET가 된다. 하나의 ILF 또는 EIF는 하나 이상의 RET를 가질 수 있다. 여러 요구사항 청크(G, Sc)의 'Target' 으로부터 도출된 데이터는 시나리오에서 정제된다. 정제된 데이터로부터 DET의 수와 RET의 수를 계산할 수 있다.

이를 수행하기 위해 본 논문에서는 데이터 기능 식별에 대해 다루는 DF1, 데이터 기능 분류에 대해 다루는 DF2, 데이터 복잡도에 대해 다루는 DF3의 3가지 규칙을 제안하였다. 규칙에 대한 정의 및 예제는 4장 사례연구를 통해 기술한다.

3.3 Step 3: 트랜잭션 기능 도출

데이터 기능인 ILF와 EIF에 대한 식별이 끝난 후에는 트랜잭션 기능인 EI, EO, EQ를 식별하여야 한다. 기능분석에서의 트랜잭션은 사용자에게 의미 있는 단위 프로세스 이면서 사용자가 식별 가능한 데이터를 입력, 출력, 조회 하는 동작을 가져야 한다. 그것은 Interaction Level에서 목표의 'Verb'와 의미상으로 대응된다. Interaction Level 목표의 'Verb'는 사용자 관점에서 대상 어플리케이션과 상호작용하기 위한 동작을 표현한다. 그러므로 'Verb'로부터 입력, 조회, 출력 동작을 도출할 수 있다. 트랜잭션 기능의 이름은 구별을 위해 'Verb + Target + (Direction)' 으로 한다. 여기서 'Direction'은 때에 따라 생략 가능하다. 예를 들어, '(Add)_{verb} (order)_{target} (to order management application)_{direction}'과 '(Add)_{verb} (order)_{target} (to warehouse management application)_{direction}'은 서로 다른 트랜잭션이다.

식별된 트랜잭션 기능은 EI, EO, EQ로 분류되어야 한다. 제안된 방법은 'Verb'의 형태를 기반으로 이를 분류한다. 목표와 시나리오에 사용되는 'Verb'는 본 논문에서 능동태로 제한된다. 또한, 본 논문은 'Verb'의 형태를 주요 의도가 데이터를 보내는 것인 'Data sending type verb'와 주요 의도가 데이터를 받는 것인 'Data receiving type verb'로 구분한다. 의미기반으로 구분된 'Verb'의 형태와 식별된 트랜잭션의 기술구조를 통해 해당 트랜잭션의 주요의도가 데이터의 입력, 출력, 조회 중 어느 것인지를 구분할 수 있다. 다음은 요구사항 문서에서 많이 사용되는 두 형태의 'verb'에 대한 예이다.

EI, EO, EQ를 식별하였으면 각 트랜잭션 기능의 복잡도를 계산하여야 한다. EI, EO, EQ의 복잡도는 FTR(File Type Reference)과 DET(Data Element Type)의 개수로 결정된다. FTR은 각 트랜잭션이 참조하는 ILF 또는 EIF 이고, DET(여기서의 DET는 데이터 기능의 DET와 다르다)는 각 트랜잭션이 참조하는 DET의 수와 에러메시지, Activation 버튼 등의 수를 합한 것이다.

이를 수행하기 위해 본 논문에서는 트랜잭션 기능 식별에 대해 다루는 TF1, 트랜잭션 기능 분류에 대해 다루는 TF2, TF3, TF4 트랜잭션 기능 복잡도에 대해 다루는 TF5, TF 6의 6가지 규칙을 제안하였다. 규칙에 대한 정의 및 예제는 4장 사례연구에서 기술한다.

4. 사례 연구 및 규칙

본 논문에서는 제안된 방법의 타당성과 적용가능성을

확인하기 위해 'Order Processing System' 예제에 제안된 방법을 적용하고 관련된 규칙을 예제와 함께 기술하였다. 'Order Processing System'은 파일창고를 가지고 있는 소규모 파일 상점을 위한 주문처리 시스템으로 수작업으로 이루어지는 주문처리를 전산화하기 위해 계획되었다. 주요 서비스는 주문정보를 관리하는 것과 창고정보를 관리하는 것이다.

본 사례연구는 목표 및 시나리오 모델링, Context elements 도출, 데이터 기능 도출, 트랜잭션 기능 도출, 미조정 기능점수 계산, 비용모델의 통한 비용 추정, 토론의 순서로 이루어진다. 자세한 내용은 세부 절에서 기술한다.

4.1 목표 및 시나리오 모델링

'Order Processing System'을 목표와 시나리오 기법으로 모델링 한 부분결과는 그림 4와 같다. 최상위 목표인 Business Level Goal은 'provide order processing service' 로 기술될 수 있다. 이를 만족하기 위한 Service Level Goal은 'provide order management service'와 'provide inventory management service'로 정제될 수 있다. 서비스 수준 목표를 만족하기 위해 상호작용 수준 목표에서는 'add order', 'add order by backorder', 'modify order', 'modify order by backorder', 'fill order', 'check inventory', 'restock inventory', 'check authority'의 목표가 있을 수 있으며 이들은 다시 각 상호작용 수준 목표를 만족시키기 위한 내부수준 목표로 분화될 수 있다.

4.2 Context elements 도출

4.1절에서 도출된 목표 및 시나리오 모델로부터 기능점수를 도출하기 위해서 먼저 컨텍스트 요소 도출 규칙인 CA1과 CA2를 적용하였다. CA1은 어플리케이션 분할에 대한 규칙이고 CA2는 에이전트 식별에 대해 다루는 규칙이다. 규칙의 내용과 적용 사례(예제)는 다음과 같다.

Context Analysis guiding rule 1 (CA 1) :

정의: 측정대상 시스템이 두 개 이상의 어플리케이션으로 분할될 경우 그 어플리케이션들은 비즈니스 목표에서 분화되며 각 어플리케이션 별로 하위 goal로 분화된다.

예제: 'Order Processing System'은 'Order management' 와 'Inventory management' 라는 두가지 서비스를 제공한다. 이는 물리적으로 분리된 두개의 다른 어

표 1 트랜잭션 기능 분류를 위한 verb 형태

Data sending type verb	Data receiving type verb
Send, display, add, dispatch, forward, direct, convey, remit, input, request, modify, fill, output, enter	Receive, get, accept, be given, pick up, collect, obtain, acquire, take, validate, check

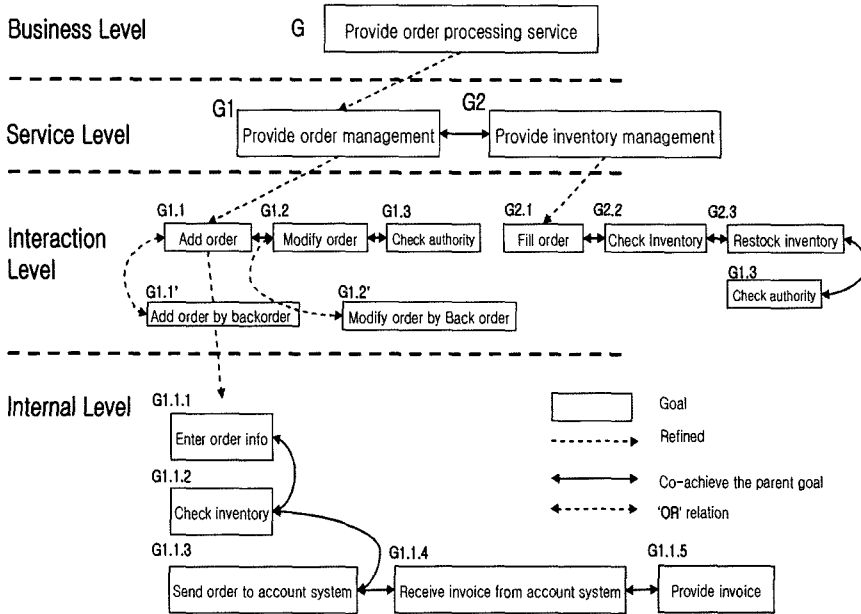


그림 4 Order Processing System의 목표 모델 일부

플리케이션으로 제공될 수 있으므로 분할이 가능하다. 분할은 비즈니스 수준 목표로부터 이루어지며 아래의 그림 5와 같이 도식화 할 수 있다.

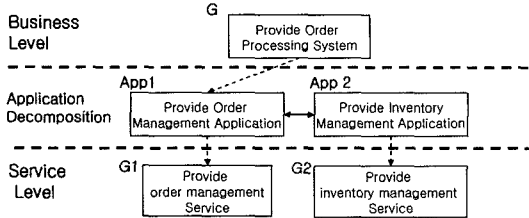


그림 5 어플리케이션의 식별과 분할

Context Analysis guiding rule 2 (CA 2)

정의: 상호작용 수준에서 'subject' 와 'direction' 에 기술된 에이전트(Agent)는 서로 상호작용한다. 에이전트 중 사람은 사용자의 후보이고 사용자와 상호작용하는 에이전트는 어플리케이션이다.

예제: 아래 예제에서 warehouse manager는 사람이므로 사용자이고, warehouse management application 은 사람과 상호작용하므로 어플리케이션이다.

(Warehouse manager)_{sub} (check)_{verb} (inventory level)_{target} (from warehouse management application)_{direction}

CA1의 적용을 통해 우리는 측정대상 어플리케이션이 두개의 어플리케이션으로 분할된다는 것을 분석하였고 이를 목표 모델과 연결할 수 있었다. 그리고CA2를 적용

하여 'Accounting Manager', 'Warehouse Manager', 'Accounting system', 'User management application' 의 에이전트를 식별할 수 있었고 이를 기반으로 그림 6 과 같은 Context diagram을 그릴 수 있었다. 사용자와 외부 어플리케이션은 측정 대상 시스템의 외부에 위치 하고 분할된 어플리케이션은 각기 다른 경계를 가지고 있다.

분할된 어플리케이션의 측정은 규칙 CA3에 따라 수행된다.

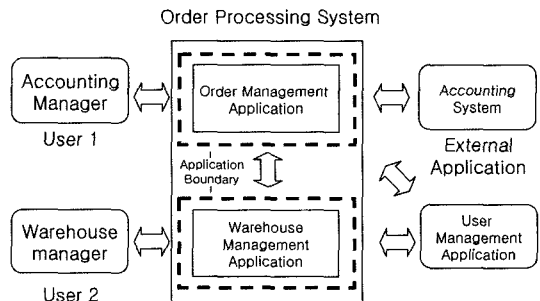


그림 6 Order Processing System의 Context Diagram

Context Analysis guiding rule 3(CA 3)

정의: 측정 대상 어플리케이션이 둘 이상일 경우, 각 어플리케이션의 기능점수는 개별적으로 측정되며, 각 어플리케이션의 측정 시, 나머지는 외부 어플리케이션으로 간주된다.

그러므로 'Order Processing System'은 두번에 걸쳐 기능점수를 측정하며 'Order management application' 측정 시에는 'inventory management application'은 외부 어플리케이션으로 간주되고 'inventory management application' 측정시에는 'Order management application'이 외부 어플리케이션으로 간주된다.

4.3 데이터 기능 도출

Context 분석을 수행한 뒤 데이터 기능 도출 규칙인 DF 1, DF 2, DF 3를 순서대로 적용하여 데이터 기능을 도출하였다. DF 1은 데이터 기능 식별에 대해 다루고, DF 2는 식별된 데이터 기능을 ILF또는 EIF로 분류하는 것에 대해 다루며 DF 3는 데이터 기능의 복잡도에 대해 다룬다. 각 규칙에 대한 정의와 예제는 아래와 같다.

Data Function Extraction Guiding Rule 1(DF 1)

정의: 상호작용 수준에서 목표의 'Target' 그리고 시나리오의 'target' 중 목표의 'target'과 연관이 없으나 사용자가 식별 가능한 'target'의 그룹은 ILF 또는 EIF의 후보이다.

제약사항: 식별된 'target'은 어플리케이션 내에서 유지되는 데이터나 제어정보이어야 한다. 'target'이 유지되는 어플리케이션은 'subject' 또는 'direction'으로부터 알 수 있다.

예제: 아래 예제 중 1은 목표, 2와 3은 해당 목표의 시나리오 중 일부이다. 1에서 'Order'는 Order Management Application에서 유지되는 논리적 데이터이다. 2와 3에서 'invoice id'와 'invoice contents'는 그룹으로 묶일 수 있고 'order'와 관련 없는 데이터 이며 외부 어플리케이션에서 유지되는 데이터이다. 그러므로 이들은 ILF 또는 EIF이다.

1. (Add)_{verb} (**Order**)_{target} (to Order Management Application)_{direction}
2. (Order Management Application)_{sub} (receive)_{verb} (**invoice id**)_{target} (from Accounting System)_{direction}
3. (Order Management Application)_{sub} (receive)_{verb} (**invoice contents**)_{target} (from Accounting System)_{direction}

예제에서와 같은 방법으로 DF 1을 'Order Processing System'의 다른 목표와 시나리오에 적용하면 'Order Management Application'에서는 'order', 'user', 'invoice', 'inventory'의 데이터 기능이 식별되고 'Warehouse Management Application'에서는 'inventory', 'user', 'order'의 데이터 기능이 식별된다.

Data Function Extraction Guiding Rule 2(DF 2)

정의: DF 1에서 식별된 ILF 또는 EIF 후보는 시나

리오에서 정제된다. 정제된 데이터가 모두 측정대상 어플리케이션에서 유지되면 식별된 데이터 기능은 측정대상 어플리케이션의 ILF 이다. 반대로 정제된 데이터가 외부 어플리케이션에서 유지되면 EIF이다.

제약사항: 데이터가 여러 목표에 나타나면 관련된 모든 목표를 검토해야 한다. 또한 데이터가 유지되는 어플리케이션에 대한 정보는 'subject'와 'direction'에 있다.

예제: 그림 7에서 목표(G2.2)의 'inventory'를 정제된 데이터들은 시나리오에 있다. 'direction'과 'subject'에서 확인할 수 있듯이 이들 모두 'warehouse management application' 내에서 유지되므로 해당 어플리케이션의 ILF 이다. 반면 DF 1의 예제에서 2와 3에서 도출된 'invoice id'와 'invoice contents'는 외부 어플리케이션인 'accounting system'에서 유지되므로 EIF이다.

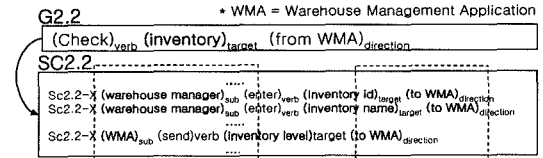


그림 7 ILF 기능 식별

예제에서와 같은 방법으로 DF 2를 다른 목표 및 시나리오에 적용하면 'Order Management Application'에서 'order'는 ILF로 분류되고 'user', 'invoice', 'inventory'는 EIF로 분류된다. 'Warehouse Management Application'에서는 'inventory'는 ILF로 분류되고 'user', 'order'는 EIF로 분류된다.

Data Function Extraction Guiding Rule 3(DF 3)

정의: ILF 또는 EIF로 식별된 데이터기능의 RET와 DET는 해당 데이터기능의 정제된 데이터로부터 산정할 수 있다. ILF 또는 EIF는 시나리오에 정제된 데이터를 가지고 있다. 정제된 데이터 각각은 DET이고 이들이 사용자에게 의미 있는 그룹으로 묶이면 그 그룹은 RET이다. RETC(RET Count)와 DETC(DET Count)는 RET와 DET의 개수를 의미하며 이를 IFPUG의 RET/DET Complexity matrix [10]에 대입하여 복잡도를 얻을 수 있다.

제약사항: 반복데이터는 하나로 간주한다. 해당 데이터를 포함하는 모든 (G, Sc)를 고려해야 한다.

예제: 그림 8에 따르면 'order'는 G1.1과 G1.2의 'Target'에 공통으로 있다. 'order'의 정제된 데이터는 customer point of contact, customer name 등의 10가지 정보가 있고 이는 customer information과 order

item information의 두 그룹으로 묶일 수 있다. 또한 DF 2에 의해서 Invoice id, invoice contents의 데이터는 invoice information으로 그룹화되어 새로운 EIF로 식별된다. 이를 기반으로 식별된 ILF와 EIF의 기본 RET와 DET를 계산할 수 있다.

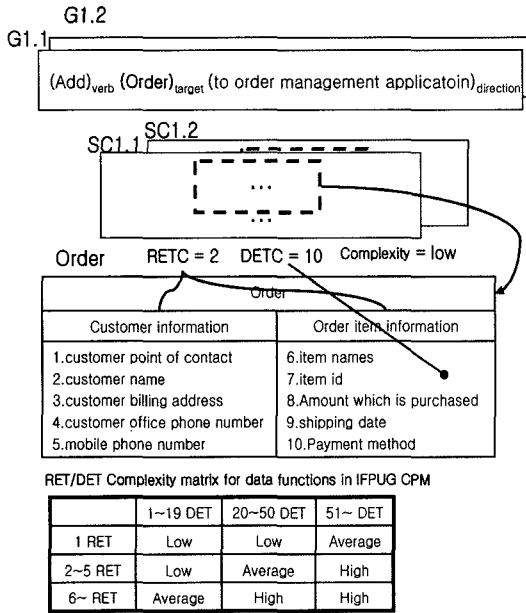


그림 8 데이터 기능의 복잡도 계산

예제에서와 같은 방법으로 DF 3을 모든 목표 및 시나리오에 적용하면 'Order Processing System'의 데이터 기능 복잡도를 결정할 수 있으며 그 결과는 표 2와 같다.

표 2 데이터 기능 도출 결과

Order Management Application				
Data F.	Function name	RET	DET	CPX.
ILF	Order	2	10	low
EIF	User	1	2	low
	Invoice	1	2	low
	inventory	1	6	low
Warehouse Management Application				
Tran.F.	Function name	RET	DET	CPX.
ILF	Inventory	1	6	low
EIF	User	1	2	low
	order	2	10	low

4.4 트랜잭션 기능 식별 및 복잡도 결정

데이터 기능 도출을 수행한 뒤 트랜잭션 기능 도출 규칙인 TF 1, TF 2, TF 3, TF 4, TF 5, TF 6를 적용하여 트랜잭션 기능을 도출하였다. TF 1은 트랜잭션

기능 식별에 대해 다루고, TF 2, TF 3, TF 4는 식별된 트랜잭션 기능을 EI, EO, EQ로 분류하는 것에 대해 다루며 TF 5, TF 6은 트랜잭션 기능의 복잡도를 low, medium, high로 결정하는 것에 대해 다룬다. 각 규칙에 대한 정의와 예제는 아래와 같다.

Transaction Function Extraction Guiding Rule 1 (TF 1)

정의: 목표의 'verb + target + (direction)'이 EI, EO, EQ후보이다. 'Direction'은 생략 가능하다. 만약 'Verb+Target+Direction'은 같고 'Way'가 다른 Goal이 있으면 이들은 하나의 트랜잭션으로 식별된다.

예제: 아래의 예제에서 1과 2는 order를 입력하는 동작을 표현하는 Interaction Goal이다. 두 Goal은 Way만 다르므로 하나의 트랜잭션'Add order to order management application'으로 식별된다.

1. (Add)_{verb} (order)_{target} (to Order Management Application)_{direction}
2. (Add)_{verb} (order)_{target} (to Order Management Application)_{direction} (through backorder)_{way}

예제에서와 같은 방법으로 TF 1을 다른 목표와 시나리오에 적용하면 'Order Management Application'에서는 'add order', 'modify order', 'check authority'의 트랜잭션 기능이 식별되고 'Warehouse Management Application'에서는 'fill order', 'Restock order', 'check inventory', 'check authority'의 트랜잭션 기능이 식별된다.

Transaction Function Extraction Guiding Rule 2 (TF 2)

정의: 트랜잭션으로 식별된 목표의 기술형태가 <(User or External Application)_{sub} + (Data sending type verb)_{verb} + (Target data)_{target} + (Measured Application)_{direction}>이거나 <(Measured Application)_{sub} + (Data receiving type verb)_{verb} + (Target data)_{target} + (User or External Application)_{direction}>이면 EI이다.

제약사항: 목표의 'Subject'는 해당 목표를 유도한 시나리오의 'subject'로 간주한다.

예제: 아래 예제에서 'subject'가 user 이고 'direction'은 측정대상 어플리케이션이다. 'Verb'는 data sending type verb로 주요의도는 order 데이터를 측정대상 어플리케이션에 추가하는 것이므로 EI이다.

(Order manager)_{sub} (add)_{verb} (order)_{target} (to order Management Application)_{direction}

Transaction Function Extraction Guiding Rule 3 (TF 3)

정의: 트랜잭션으로 기능으로 식별된 목표의 기술형태가 $\langle (User\ or\ External\ Application)_{sub} + (Data\ receiving\ type\ verb)_{verb} + (Target\ data)_{Target} + (Measured\ Application)_{direction} \rangle$ 이거나 $\langle (Measured\ Application)_{sub} + (Data\ sending\ type\ verb)_{verb} + (Target\ data)_{Target} + (User\ or\ External\ Application)_{direction} \rangle$ 이면 EO 또는 EQ 이다.

예제: 아래 예제에서 'subject'가 user 이고 'direction'은 측정대상 어플리케이션이다. 예제의 'Verb'는 data receiving type verb로 주요 의도는 inventory information을 전송 받아 확인하는 것이므로 EQ 또는 EO의 후보이다.

$(Warehouse\ manager)_{sub} (check)_{verb} (inventory\ information)_{target} (from\ warehouse\ management\ application)_{direction}$

Transaction Function Extraction Guiding Rule 4 (TF 4)

정의: EI, EO 후보 중 해당 기능이 수학계산, 파생 데이터, 데이터 기능의 생성/수정 삭제, 시스템 동작 변경을 포함할 경우 EO 이고 그렇지 않은 경우는 EQ 이다.

예제: 아래 예제는 TF 3을 통해서 EO 또는 EQ의 후보로 식별되었다. 이 트랜잭션의 주요 의도는 inventory information을 전송 받아 조회하는 것으로, 수학공식 파생데이터, 데이터 기능의 생성/수정/삭제, 시스템 동작변

경을 포함하고 있지 않다. 그러므로 EQ이다.

$(Warehouse\ manager)_{sub} (check)_{verb} (inventory\ information)_{target} (from\ order\ management\ application)_{direction}$

예제에서와 같이 TF 2, TF3, TF4 를 다른 목표와 시나리오에 적용하면 TF 1을 통해 식별된 트랜잭션 기능들을 EI, EO, EQ분류할 수 있다. 'Order Management Application'에서 'order'는 ILF로 분류되고 'user', 'invoice', 'inventory'는 EIF로 분류된다. 'Warehouse Management Application'에서 'inventory'는 ILF로 분류되고 'user', 'order' 는 EIF로 분류된다.

Transaction Function Extraction Guiding Rule 5 (TF 5)

정의: 트랜잭션으로 기능의 복잡도 결정을 위한 DETC(DET Count)는 시나리오의 'target'으로부터 도출되며 아래 요소들의 합이다. (식별된 트랜잭션기능을 완수하기 위한 시나리오의 'target'에 존재하는 반복되지 않는 정제 데이터 수) + 1(에러, 완료, 확인 메시지) + 1(기능 기동 방법) + 1(EO, EQ를 위한 검색 또는 입력)

제약사항 측정대상 시스템에 의해 파생되거나, 대상 시스템 경계를 넘지 않는 데이터는 DET로 측정하지 않는다. 에러/확인/완료 메시지는 1개로 측정한다. 기능 기동 방법(예를 들어 버튼)도 1개로 측정한다.

예제: 그림 9는 EI로 식별된 'add order' 에 대한 DETC 측정 예제이다. 'Add order'의 시나리오에는 'order'와 관련된 데이터 10개, 'Invoice'와 관련된 데이

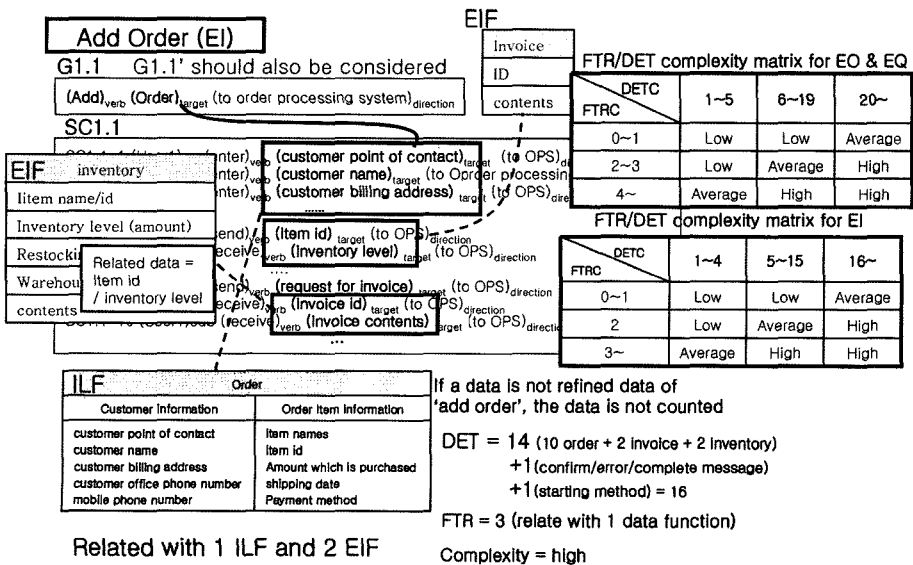


그림 9 트랜잭션 기능 복잡도 결정

타 2개, 'Inventory'와 관련된 데이터 2개가 포함되어 있다. 여기에 확인/에러/완료 메시지와 기동방법을 추가하여 16로 측정한다.

Transaction Function Extraction Guiding Rule 6 (TF 6)

정의: 트랜잭션 기능의 복잡도 결정을 위한 FTRC (FTR Count)는 관련된 ILF 또는 EIF의 숫자이다. 관련여부는 트랜잭션 기능과 데이터 기능의 공유하는 DET 존재여부로 결정한다.

예제: 그림 9에서 'add order' 는 ILF인 'Order', EIF인 'Invoice', 'Inventory'와 관련되므로 총 3개의 데이터 기능과 관련되어 FTR은 3이다.

트랜잭션 기능의 FTRC과 DETC를 산정한 후 IFPUG에서 제공하는 FTR/DET matrix(그림 8) [10]를 통해 복잡도를 결정할 수 있다.

예제에서와 같은 방법으로 TF 5 및 TF 6을 모든 목표 및 시나리오에 적용하면 'Order Processing System'의 트랜잭션 기능 복잡도를 결정할 수 있으며 그 결과는 표 3과 같다.

표 3 트랜잭션 기능 도출 결과

Order Management Application				
Data F.	Function name	FTR	DET	CPX.
EI	Add order	3	16	high
	Modify order	3	16	high
EQ	Check authority	1	3	low
Warehouse Management Application				
Tran.F.	Function name	FTR	DET	CPX.
EI	Fill order	1	12	low
	Restock order	1	8	low
EQ	Check inventory	2	9	average
	Check authority	1	3	low

4.5 미조정 기능점수 계산

각 어플리케이션 별로 EI, EO, EQ, ILF, EIF를 식별하고 복잡도를 결정하였으면, 표 5에서와 같이 IFPUG에서 제공하는 변환 표를 이용하여 미조정 기능점수(UFP: Unadjusted Function Point)를 구할 수 있다 [10]. 예를 들어 'add order'는 EI이고 복잡도가 'high'이므로 표 5에 의해 6 UFP로 산정된다. 미조정 기능점수에 범용 시스템 특성에 근거한 조정인자를 곱하면 조정 기능점수(AFP: Adjusted Function Point)가 구해진다. 하지만 본 논문에서는 미조정 기능점수까지만을 다룬다. 그 이유는 미조정 기능점수가 비용 모델에서 보다 널리 입력으로 사용되고 있으며 국제 표준 및 정보통신부 표준에서는 미조정 기능점수까지만 채택하였기 때문이다.

'Order Processing System'의 미조정 기능점수 측정 결과는 표 4와 같다. 각 기능들의 복잡도는 그림 8과 그림 9에 있는 복잡도 변환 표에 의해 결정되었으며 최종 UFP는 표 5에 의해 결정되었다.

따라서, 'Order Management Application'는 37 UFP, 'Warehouse Management Application'은 30 UFP가 산정되었으며 'Order Processing System'는 67 UFP로 산정되었다.

표 4 Order Processing System의 미조정 기능점수

Order Management Application					
Data F.	Function name	RET /FTR	DET	CPX.	UFP
ILF	Order	2	10	low	7
EIF	User	1	2	low	5
	Invoice	1	2	low	5
	inventory	1	6	low	5
EI	Add order	3	16	high	6
	Modify order	3	16	high	6
EQ	Check authority	1	3	low	3
Subtotal					37
Warehouse Management Application					
Tran.F.	Function name	RET /FTR	DET	CPX.	UFP
ILF	Inventory	1	6	low	7
EIF	User	1	2	low	5
	order	2	10	low	5
EI	Fill order	1	12	low	3
	Restock order	1	8	low	3
EQ	Check inventory	2	9	average	4
	Check authority	1	3	low	3
Subtotal					30
Total UFP					67

표 5 복잡도/미조정기능점수 변환 표

Func. \ CPX.	Low	Average	High
EI & EQ	3	4	6
EO	4	5	6
ILF	7	10	15
EIF	5	7	10

4.6 비용모델을 이용한 비용추정

본 논문에서 제안하는 방법은 목표 및 시나리오 기반 요구사항으로부터 기능점수를 도출하는 것까지만 다룬다. 제안된 방법을 통해 도출된 기능점수는 기존의 비용 모델에 입력하여 비용 추정에 사용될 수 있다. 그러므로 제안된 방법을 통해 요구사항과 비용간의 추적관리가 가능하다.

요구사항과 비용과의 연결성을 보여주기 위해 본 논문에서는 'Order Processing System' 예제의 기능점수 측정 결과를 잘 알려진 상용 모델 중 하나인 SEER (<http://galorath.com>) 모델에 입력하여 비용추정을 해 보았다. SEER는 미국방부의 표준 비용추정 모델로 채택된 도구 중의 하나이며 널리 쓰이는 상용 모델 중 하나이다. 본 논문에서는 요구사항과 비용의 연결을 보여주는 것이 목적이므로 단순히 도구에 사용된 예제의 결과를 입력하여 결과를 도출하였다.

SEER를 이용하여 비용을 추정하기 위해서는 기능점수 이외에 SEER 모델에 특화된 입력 값을 추가로 입력하여야 한다. 본 논문에서는 'Knowledge Base'라고 하는 SEER 모델에 특화된 기능을 사용하여 5가지 시스템 특성을 입력하였다. (Platform=Business and Non-critical MIS, Application = "Transaction Processing", Acquisition Method=new development, Development Method = OOD and OOP) 이에 대한 자세한 설명은 본 논문의 범위가 아니므로 다루지 않기로 한다.

'Order Processing System' 예제의 기능점수 결과를 SEER 모델에 입력하여 얻은 결과는 그림 10 과 11에 나타나 있다.

비용추정 결과는 전체 시스템 개발에 7.77 Effort Month(Man Month)가 소요되고, Order Management 는 4.23 Effort Month, Warehouse Management는

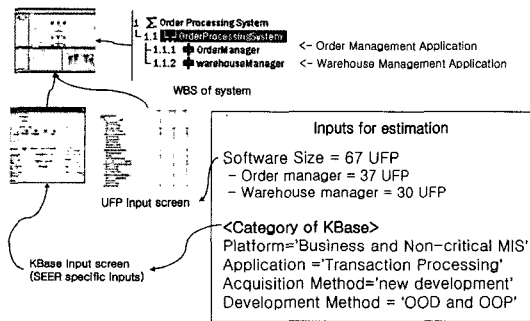


그림 10 비용추정의 입력

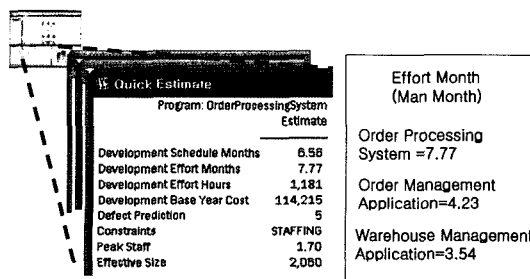


그림 11 비용추정 결과

3.54 Effort Month 소요되는 것으로 추정되었다. 모델에 입력한 평균 노임단가에 따라 개발 비는 달라지며 위의 경우는 \$ 14,700의 평균 노임단가(미국 평균 노임 단가)를 적용하여 \$ 114,215의 개발비를 추정하였다.

4.7 비용과 요구사항의 추적성 개선 평가

제안된 방안은 목표 및 시나리오 기반 요구사항으로부터 기능점수를 체계적으로 도출하는 방안을 통해 변경영향분석, 도출분석, 범주 분석 등의 추적성 분석 활동을 보다 쉽게 할 수 있도록 지원한다. 변경영향 분석은 요구사항 변경에 따른 비용변화를 분석하는 것이고, 도출분석은 비용의 도출근거를 분석하는 것이며 범주분석은 비용의 영향범주를 분석하는 것이다. 이러한 추적성 관리활동을 제안된 방법과 기존방법으로 수행할 경우의 비교는 아래 표와 같다.

표 6 추적성 분석 활동 비교

추적관리 활동	제안된 방법	기존 방법
변경 영향 분석	변경된 목표와 연결된 상호작용 수준 목표에 대해 제안된 규칙을 통해 다시 계산	기능점수 전문가에게 의뢰하여야 함
도출분석	기능점수가 각 수준별 목표에 할당되므로 목표 모델의 연결을 통해 도출분석 가능	기능점수 전문가에게 의뢰하여야 함
범주 분석	특정 서비스나 기능이 어느 정도 기능점수에 기여하는지는 하위 수준 목표의 기능점수 할당의 합으로 알 수 있음	기능점수 전문가에게 의뢰하여야 함

기존의 방법으로 추적성 관리를 하기 위해서는 기능점수 전문가의 도움을 받아야 하므로 의사소통 문제, 시간관리의 비효율성 문제 등이 존재한다 하지만 제안된 방법을 사용하면 요구사항 엔지니어 또는 기능점수 분석가 혼자서 보다 체계적인 관리를 할 수 있다.

본 논문에서 제안한 'Order Processing System' 예제를 통해 이를 설명하면 다음과 같다. 그림 12는 'Order Processing System'의 목표 모델에 산정된 기능점수를 할당 한 결과를 보여준다.

변경 영향분석을 제안된 방법으로 수행하기 위해서는 변경된 목표에 대해 제안된 규칙을 다시 적용하여 계산하면 된다. 예를들어 요구사항이 변경되어 상호작용 수준 목표인 'Check authority'를 제거해야 하고 이에 대한 비용영향을 분석해야 한다면, 'Check authority'와 이와 연결된 EIF인 'user'를 같이 고려하여야 한다. 이들 둘이 'Order Management Application'과 'Inventory Management application'에 같이 존재하므로 두 어플리케이션에서 각 8 UFP 씩 총 16 UFP가 감소함을 알 수 있다.

도출분석이나 범주분석은 추상화 수준별 목표모델을 추적함으로써 수행될 수 있다. 예를들어 'Fill order'라는 EI 기능이 어떤 서비스로부터 도출되었는지를 분석하기 위해서는 그림 12의 추적연계설정을 추적하면 'Inventory Management Service'로부터 도출되었음을 알 수 있다. 또한 'order management service'와 'inventory management service'가 전체에서 차지하는 비용영향은 각 서비스 수준 목표가 37 UFP와 30 UFP로 측정되므로 'Order management Service'가 7 UFP만큼 더 영향을 준다는 것을 알 수 있다. 각 서비스를 만족하는 상호작용 수준 목표가 어느정도의 비용기여를 하는지도 그림 12를 통해 알 수 있다.

5. 결론 및 향후연구

요구사항과 비용은 밀접한 관계를 가지고 있다. 요구사항 변경에 따라 비용 추정 결과도 같이 바뀌고 비용 추정 결과는 다시 요구사항에 영향을 준다.

본 논문에서 제안하는 방법은 목표 및 시나리오 기반의 요구사항으로부터 체계적으로 기능점수를 도출하기 위한 규칙을 제공한다. 그 규칙은 목표 및 시나리오 기반 요구사항과 기능점수 간의 연결을 명확히 하며 추적성 관리의 기반이 된다. 그러므로 각 추상화 수준에서의 요구사항 변경에 따른 비용영향 분석이 보다 효과적으로 수행될 수 있다.

제안된 기법은 Order Processing System 예제에 적용되었으며 67 UFP를 도출하였다. 또한, 제안된 방법은 다양한 프로젝트에의 적용가능성을 확인하기 위해 'Course

Registration System' 예제에도 적용되어 77 UFP를 얻었으나, 지면관계상 이를 포함하지 않았다.

제안된 방법의 한계는 IFPUG의 기능점수를 다루기 때문에 IFPUG의 기능점수가 취약한 내장형 시스템에는 적합하지 않다는 점이다. 하지만 기능점수의 적용이 많이 이루어지는 사용자 데이터 처리를 다루는 도메인에는 적용이 가능하다. 또한, 제안된 규칙은 지원도구와 결합되어야 활용도가 높아지고 요구사항 분석과 기능분석의 효율적인 통합수행이 가능하다. 우리는 지원도구의 개발을 계획 중에 있으며 이를 향후 연구로 남겨놓는다. 또한, 보다 많은 프로젝트에 적용하여 제안된 방법의 실효성을 검증하는 것과 객체지향 기법에 적용될 수 있는 Use Case Point의 도출방법에 대한 연구도 계획 중에 있다.

참고 문헌

- [1] V. Ambriola, V. Gervasi, "Processing natural language requirements," Proceedings of the 12th international conference on Automated software engineering pp 36, IEEE Computer Society, 1997.
- [2] Boehm, B., Hoh In, "Software Cost Option Strategy Tool (S-COST)," Proceedings of 20th International Computer Software and Applications Conference (COMPSAC '96) pp15-20, Aug. 1996.
- [3] Karlsson, J., Ryan, K, "A cost-value approach for prioritizing requirements," IEEE Software Volume 14, Issue 5 pp67-74, Sept.-Oct. 1997.
- [4] Boehm, B., Software Engineering Economics, Prentice-Hall, 1981.

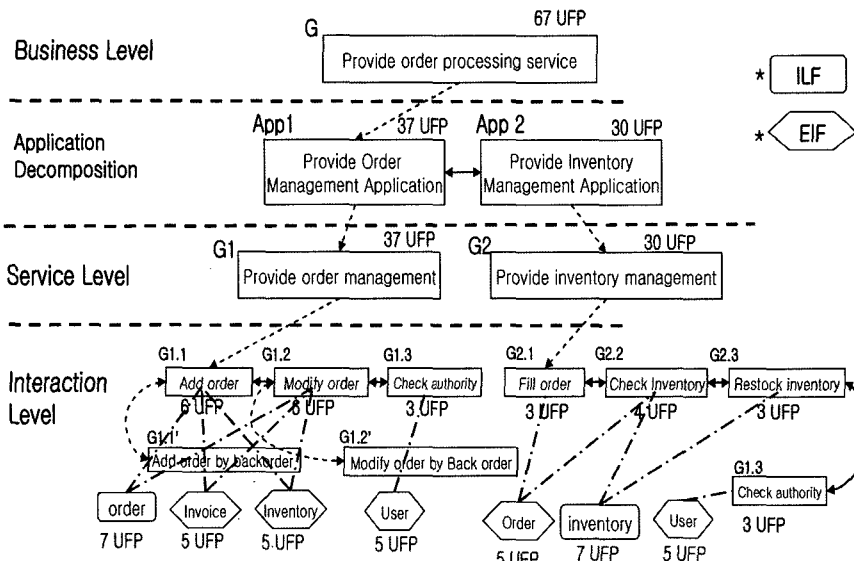


그림 12 목표 및 시나리오 기반 요구사항에의 기능점수 할당

- [5] Albrecht, A. J., "Measuring Application Development Productivity," IBM Applications Development Symposium, Monterey CA, 1979.
- [6] David Garmus, David Herron, Function Point Analysis, Addison Wesley, 2001.
- [7] ISO-IEC, International ISO/IEC Standard 14143-1, Information Technology - Software measurement - Functional size, Part 1: Definition of Concepts, 1998.
- [8] Takuya Uemura, Shinji Kusumoto, Katsuro Inoue, "Function Point Measurement Tool for UML Design Specification," IEEE METRICS 1999: 62-., 1999.
- [9] 정보통신부, 소프트웨어 사업대가의 기준, 정통부 고시 제 2004-8호, 2004.
- [10] IFPUG, Function Point Counting Practice Manual Release 4.1.1, International Function Point User Group, 2000.
- [11] Hwasil Yang, Minseong Kim, Sooyong Park, Vijayan Sugumaran, "A Process and Tool Support for Managing Activity and Resource Conflicts Based on Requirements Classification," NLDB 2005, LNCS 3513, pp. 114-125, 2005. 6.
- [12] A. Dardenne, A. van Lamswerde and S. Fickas, "Goal-Directed Requirements Acquisition," Science of Computer Programming, Vol. 20, 1993, pp. 3-50.
- [13] C. Rolland, C. Souveyet, and C. Ben Achour, "Guiding goal modeling using scenarios," IEEE Transactions on Software Engineering, Vol. 24, Dec. 1998, pp.1055-1071.
- [14] Jintae Kim, Sooyong Park, Vijayan Sugumaran, "A Linguistics-Based Approach for Use Case Driven Analysis Using Goal and Scenario Authoring," NLDB 2004, LNCS 3136, Springer-Verlag, pp. 159-170, 2004. 6.



한 지 영
 2005년 서강대학교 컴퓨터학과(공학사)
 2005년~현재 서강대학교 컴퓨터학과 석사과정. 관심분야는 요구공학, 소프트웨어 프로세스, 아키텍처



최 순 황
 2001년 서강대학교 컴퓨터학(공학사)
 2003년 서강대학교 컴퓨터학(공학석사)
 2003년~현재 서강대학교 컴퓨터학 박사과정. 관심분야는 요구공학, 프로덕트 라인, 소프트웨어 프로세스, 소프트웨어 아키텍처

김 진 태
 정보과학회논문지 : 소프트웨어 및 응용
 제 33 권 제 1 호 참조

박 수 용
 정보과학회논문지 : 소프트웨어 및 응용
 제 33 권 제 1 호 참조