

요구분석을 위한 Event Process Modeling

김영보

Event Process Modeling for Requirements Analysis

Kim, Young Bo

E-mail : tonextday@paran.com

요 약

시스템 개발의 성공과 실패는 요구사항 정의의 명확함에 따라 좌우된다고 해도 지나침이 없다. 요구분석은 요구분석을 수행하는 프로세스 측면과 요구사항을 정의하는 방법적인 측면으로 나눌 수 있다. 프로세스 측면은 점증적으로 요구사항을 명확하게 정의하기 위한 것이며, 방법적인 측면은 논리와 근거를 갖춘 형태로 요구사항을 정의하기 위한 것이다. 즉, 요구분석 단계와 요구사항 정의 방법이 요구된다.

본 논문에서는 이벤트 프로세스 모델링의 요구분석 5단계와 요구사항 변경을 예방하는 방법을 제시한다. 요구분석 완료 기준을 제시하고, 이벤트와 프로세스 방법으로 요구사항을 정의하는 방법을 설명한다.

1. 서론

포괄적으로 제시된 요구사항을 한번에 명세화할 수 없으므로 요구사항 이해, 정의, 분석, 확정하는 단계를 점증적으로 수행해야 한다. 요구사항 이해는 현행 비즈니스 프로세스를 조사하고 현행 문제점을 추출하는 단계이며, 요구사항 정의는 목표 비즈니스 프로세스를 수립하고, 각각의 비즈니스 프로세스를 유스케이스로 변환한 후, 사용자 관점에서 요구사항을 정의하는 단계이다. 요구사항 분석은 시스템 처리 관점에서 요구사항을 정의하는 단계이며, 요구사항 확정은 프로토타입을 작성하여 요구사항 반영 여부를 검토하고 요구사항을 확정하는 단계이다.

행동의 원인이 되는 이벤트가 있어야 행동을 시작하게 되듯이, 이벤트를 발생시켜 다른 프로세스

가 수행하게 되는 원인을 제공해야 한다. 또한, 발생한 이벤트를 받아서 처리한 프로세스가 다시 이벤트를 발생시켜 다음 프로세스가 수행하게 되는 원인을 제공한다. 이와 같이 이벤트와 프로세스가 서로 연결된 시나리오 흐름이 되어야 근거와 논리를 갖춘 형태로 요구사항을 정의한 것이다.

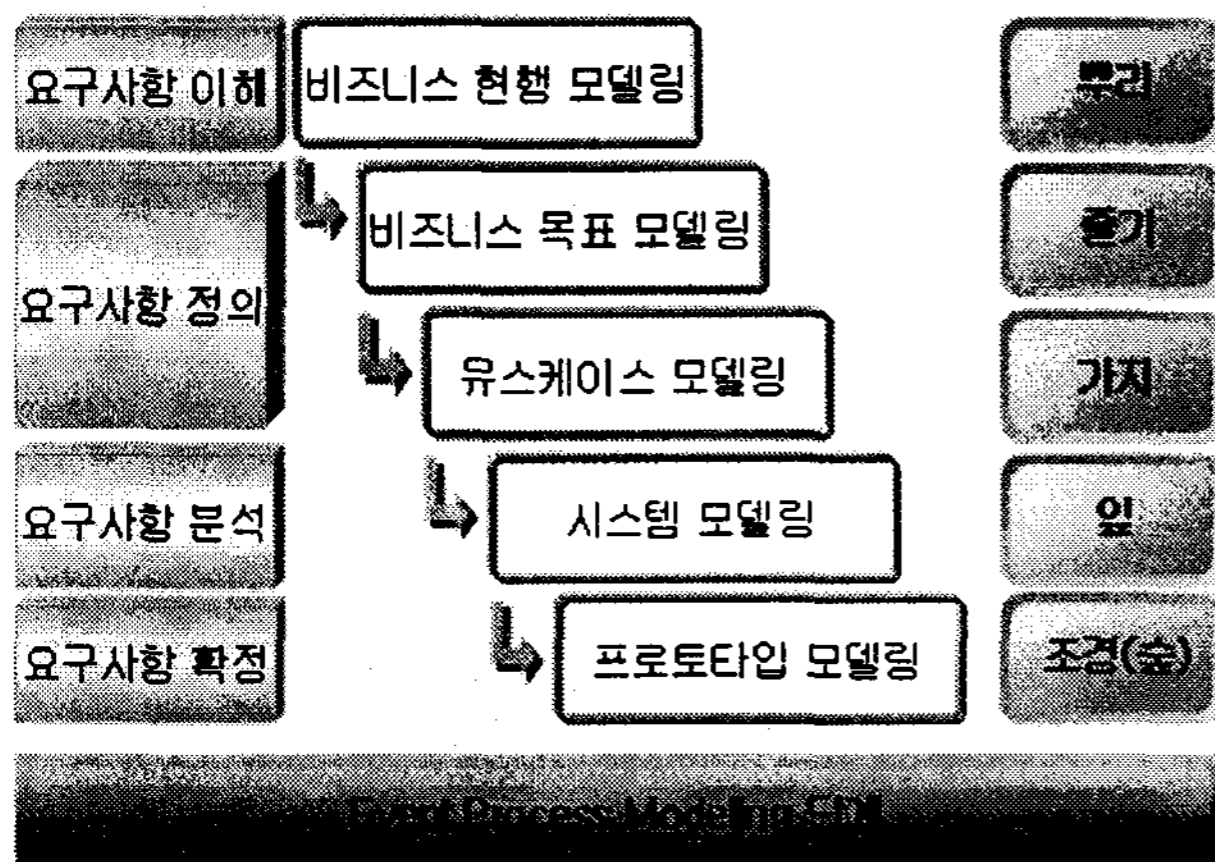
문서로 작성하여 요구사항을 확정하였다면 요구사항 변경이 발생한다. 이것은 그 누구의 잘못이 아니라 문서가 갖는 한계 때문이다. 그러므로 문서로 작성한 요구사항을 구체화, 형상화시켜야 한다. 즉, 프로토타입을 작성해야 하며, 프로토타입은 변경 사항이 없다면 실제 사용하게 될 메뉴/화면/보고서 형태로 면밀하게 작성한다. 작성한 프로토타입을 사용자가 만족할 때까지 검토와 수정을 반복한다. 면밀하게 작성한 프로토타입을 확정

하게 되면 요구 사항, 시스템 기능, 시스템 범위를 확정하게 된다. 이것이 요구분석을 완료한 것이다.

2. 본론

2.1 요구분석 5단계

이벤트 프로세스 모델링은 5단계로 나누어 요구사항 이해, 정의, 분석, 확정을 수행한다. 나무는 뿌리, 줄기, 가지, 잎을 가지고 있다. 비즈니스 현행 모델링은 뿌리를 만드는 단계이고, 비즈니스 목표 모델링은 줄기를 만드는 단계이다. 유스케이스 모델링은 가지를 만드는 단계이고, 시스템 모델링은 잎을 만드는 단계이다. 마지막으로 프로토타입 모델링은 아름다운 숲이 되도록 조경을 하는 단계이다.



[그림1] 이벤트 프로세스 모델링 단계

2.1.1 비즈니스 현행 모델링

비즈니스 현행 모델링은 현행 비즈니스 프로세스를 조사하여 현행 비즈니스 프로세스를 이해하고 문제점을 추출하는 단계이다. 시스템 개발은 사용자의 요구사항을 시스템으로 구축하는 것이다. 그러므로 사용자의 요구사항을 정확하게 이해하는 것은 기본적인 사항이다. 사용자는 문제점을 해결하기 위해 요구사항을 제시한다. 문제점에 대한 해결방안을 수립하기 위해서는 사용자가 현재 처리하고 있는 비즈니스 프로세스를 이해해야 한다.

이것은 사용자가 제시한 요구사항의 근본적인 원인과 목적을 정확하게 이해하기 위한 것이다.

많은 분석가들이 현행 비즈니스 프로세스 조사를 소홀히 하는 경향이 있는데, 이것은 위험한 접근 방법이다. 현재 상황을 정확하게 모르면 문제점에 대한 해결방안을 도출할 수 없을 뿐만 아니라, 현행 비즈니스 프로세스를 이해하지 못한 상태에서 요구사항을 정의한다는 것은 모래 위에 집을 짓는 것과 다를 없다.

문제점이 추출되면 해결방안을 찾아야 한다고 생각하는 것은 당연하다. 하지만, 다음 단계인 비즈니스 목표 모델링 단계로 넘겨야 한다. 왜냐하면, 문제점에 대한 해결방안은 시스템 전체적인 관점에서 수립해야 하며, 문제점에 대한 해결방안을 찾는데 몰두하게 되면 정작 현행 비즈니스 프로세스를 이해하는 것과 문제점을 추출하는 것에 소홀해지기 때문이다.

현행 비즈니스 프로세스를 조사하다 보면 사용자가 요구사항을 거론하게 되는데, 그 때에는 회의록에 기록하는 정도로 끝내야 한다. 왜냐하면, 현행 비즈니스 프로세스 전체에 대한 이해를 바탕으로 요구사항을 정의해야 하는데, 아직 현행 비즈니스 프로세스 조사가 끝나지 않았기 때문이다.

2.1.2 비즈니스 목표 모델링

비즈니스 목표 모델링은 목표 비즈니스 프로세스를 정의하는 단계이다. 비즈니스 목표 모델링은 BPR(Business Process Reengineering) 관점에서 목표 비즈니스 프로세스를 정의하는 것이 아니라, 요구사항을 정의하는 관점에서 목표 비즈니스 프로세스를 정의한다.

목표 비즈니스 프로세스는 비즈니스 프로세스 단위로 요구사항을 정의한다. 목표 비즈니스 프로세스는 시스템으로 처리하는 온라인(On-Line) 비즈니스 프로세스뿐만 아니라 '결재를 받는 것'과 같이 오프라인(Off-Line)에서 수행하는 비즈니스 프로세스도 포함해야 한다. 그러므로 오프라인 비

즈니스 프로세스도 요구사항으로 간주하여 목표 비즈니스 프로세스를 정의한다.

요구사항 목표는 적어도 몇 개의 비즈니스 프로세스가 결합하여 수행해야 달성할 수 있다. 비즈니스 프로세스는 흐름을 가지고 있으며, 이것을 비즈니스 시나리오(Scenario)라고 한다. 즉, 비즈니스 목표 모델링은 요구사항 목표를 달성할 수 있도록 비즈니스 시나리오를 정의하는 단계이다.

비즈니스 현행 모델링 단계에서 추출된 문제점에 대한 해결방안을 수립하여 목표 비즈니스 프로세스에 반영한다. 문제점에 대한 해결방안은 시스템 처리를 동반해야 하며, 시스템 처리를 동반하지 않는 문제점은 요구사항 범주에 속하지 않는다. 즉, 오프라인 비즈니스 프로세스가 잘못되어 발생하는 문제점은 BPR 관점의 비즈니스 모델링 범주에 속한다. 하지만, 오프라인 비즈니스 프로세스가 요구사항에 속하는 것은 문제점 해결방안 수립과는 다른 관점이다. 문제점을 해결하지 않는 것이 요구사항에서 제외시키는 것은 아니다.

사용자가 제시한 요구사항에는 틀림없이 목표가 있다. 사용자가 인원과 시간을 투입해 시스템을 개발하려는 것은 시스템 처리를 통해 사용자의 목표를 달성하기 위한 것이다. 그러므로 사용자가 명확하게 요구사항을 제시해야 하는데, 사용자는 명확하게 요구사항을 제시하지 못한다. '사용자가 요구사항 제시하지 못한다'는 것은 요구사항 목표를 제시하지 못하는 것이 아니라, 요구사항 목표를 달성하기 위해 무엇을 할 것인가에 대한 상세한 사항을 제시하지 못하는 것이다.

사용자 입장에서 생각해보면 요구사항을 어떻게 제시해야 명확한 것인가라는 자체를 모르기 때문에 요구사항을 명확하게 제시할 수 없다. 따라서 처음부터 사용자가 요구사항을 명확하게 제시하지 못할 것이라고 생각하고 접근해야 한다. 이것이 요구사항을 추출하는 기본적인 접근 방법이다. 막연히 사용자가 요구사항을 제시해야 한다는 것은 요구사항을 추출하기 위한 접근 방법이 아니다.

최종적으로 사용자가 시스템을 사용하게 되므로 사용자가 주체가 되어 시스템을 수행할 수 있도록 해야 한다. 시스템이 주체가 되어 사용자가 따라가면서 시스템을 사용할 수는 없다. 많은 개발자들이 살아 숨쉬는 생명력 있는 시스템을 개발하지 못하는 것은 시스템 중심의 시스템을 개발하기 때문이다. 사용자가 주체가 되어 시스템을 수행하기 위해서는 사용자의 행동을 중심으로 요구사항을 정의해야 한다. 시스템이 사용자의 행동을 받아 처리하는 형태가 되어야 한다.

다음 사례는 '추천 의뢰 접수' 비즈니스 시나리오를 비즈니스 목표 모델로 작성한 것이며, [그림2]는 이에 대한 이벤트 프로세스 다이어그램이다.

비즈니스 목표 모델

시스템 명칭	취업관리 시스템
모델 명칭	추천 의뢰 접수

■ 개요 및 목표

1 기업의 추천의뢰서를 시스템에 등록하여 관리한다. 추천의뢰는 홈페이지에서 할 수 있으며, 이것은 '홈페이지 접수' 모델에 정의한다.

■ 선행 조건

- 1 취업담당자가 기업에 추천의뢰서를 발송하였다.
 - 1.1 {조건} 추천의뢰서 작성일 기준으로 과거 5년 동안 추천의뢰를 한번이라도 하였던 기업이 대상이 된다.
 - 1.2 {설명} 추천의뢰서를 받은 기업만 추천의뢰를 할 수 있는 것은 아니다.

■ 시작 이벤트

1 기업이 추천의뢰를 신청한다.

■ 모델 시나리오

1 취업담당자가 기업으로부터 추천의뢰서를 접수한다.

- 1.1 전화로 접수한다. : [RV1]
- 1.2 전자메일로 접수한다. : [RV2]
- 1.3 FAX로 접수한다. : [RV3]
- 1.4 우편으로 접수한다. : [RV3]
- 2 취업담당자는 접수된 추천의뢰서를 검토한다. : [CF1]
- 3 취업담당자는 접수된 추천의뢰서를 입력한다. : [CF2]
- 4 취업담당자는 추천의뢰 접수현황을 출력한다.
- 5 취업담당자는 추천의뢰 접수현황을 취업관리자에게 결재를 받는다.
- 6 취업관리자는 추천교수를 입력한다. : [CF3]

■ 이벤트 처리

[RV1]

- 1 취업담당자는 기업에게 추천의뢰서 양식을 전자메일로 발송한다.

1.1 {규칙} 추천의뢰는 전화로 할 수 없다.

[RV2]

- 1 취업담당자는 수신한 전자메일을 출력한다.
- 2 취업담당자는 출력한 전자메일을 추천의뢰 접수대장에 철한다.

[RV3]

- 1 취업담당자는 출력한 전자메일을 추천의뢰 접수대장에 철한다.

[CF1]

- 1 필수 기재 항목이 누락되었다.
 - 1.1 취업담당자는 추천의뢰 기업에 문의하여 누락된 항목을 추천의뢰서에 기재한다.

[CF2]

- 1 시스템은 입력한 추천의뢰 내역을 저장한다.
 - 1.1 시스템은 추천의뢰번호를 부여한다.
- 2 취업담당자는 시스템이 부여한 추천의뢰번호를 추천의뢰서에 기입한다.

[CF3]

- 1 시스템은 추천교수에게 추천요청 메일을 발송한다.

■ 비즈니스 규칙

- 1 BS001: 기업은 전화로 추천의뢰를 할 수 없다.

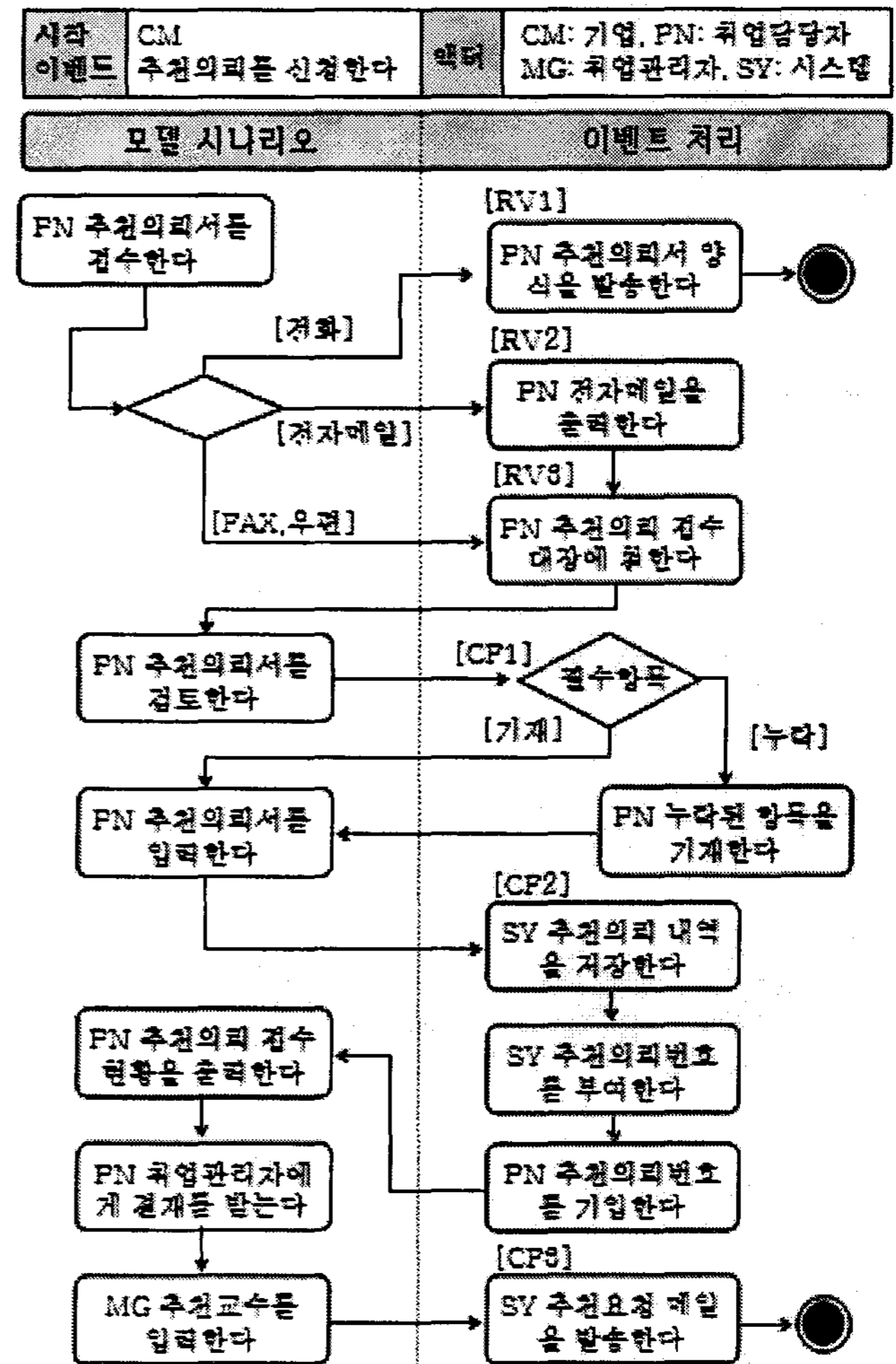
■ 참고 자료

자료번호	자료명칭
101	추천 의뢰서
102	추천의뢰 현황

문제점 및 해결방안

번호	구분	문제점 및 해결방안
1	문제점	추천의뢰 정보를 문서로 관리하고 있어 정보를 활용하는데 한계가 있다.
	해결방안	추천의뢰서를 시스템에 입력하여 관리한다

이벤트 프로세스 다이어그램



[그림2] 이벤트 프로세스 다이어그램

2.1.3 유스케이스 모델링

유스케이스 모델링은 요구사항 목표를 달성하기 위해 액터(Actor)와 시스템이 수행할 것을 사용자 관점에서 정의하는 단계이다. 액터가 행동을 하면 시스템은 액터의 행동을 받아서 처리하고, 시스템 처리 결과는 액터가 행동하게 되는 동기를 제공하여 액터가 행동하게 된다. 요구사항 목표는 액터와 시스템이 행동하는 일련의 이벤트와 프로세스 흐름을 통하여 달성하게 된다. 즉, 유스케이스 모델링은 요구사항 목표를 달성하도록 유스케이스 시나리오를 정의하는 단계이다.

비즈니스 시나리오에 속한 비즈니스 프로세스가 유스케이스가 된다는 것은 유스케이스는 한정적인 목표를 갖는 것을 의미한다. 인터넷 쇼핑몰에서 회원가입을 하고 상품을 구매하는 비즈니스 시나리오에서 회원가입 유스케이스는 회원가입이라는 한정된 목표를 갖는다. 회원가입 유스케이스가 목표를 달성하면 다른 유스케이스가 달성된 목표를 받아서 수행하는 일련의 흐름은 비즈니스 시나리오에 정의된 상태이므로 회원가입 유스케이스는 자신의 목표를 달성할 수 있도록 시나리오를 정의하면 된다.



[그림3] 유스케이스 접근 방법

다양한 환경에서 제시되는 요구사항을 완전하게 유스케이스에 정의하기 위해서는 사용자 관점, 액터의 행동 관점, 시스템 처리 관점, 시나리오 관점으로 접근해야 한다. 유스케이스에 접근하는 방법은 비즈니스 시나리오에 접근하는 방법이기도 하다. 요구사항의 크기와 범위에 차이가 있는 것이지만 기본적인 접근 방법은 동일하다.

사용자 관점이란 사용자 입장에서 요구사항을 정의하는 것이다. 사용자는 시스템 처리 방법과 어려움에는 관심이 없으며, 쉽고 편리하게 시스템을 사용하여 자신의 목적을 달성하는 것에 관심을 갖고 있다. 사용자는 시스템을 사용함에 있어 자신이 해야 할 것만 생각하고, 시스템이 처리할 것은 생각하지 않는다. 사용자는 자신이 시스템을 사용한다는 입장에서 요구사항을 제시한다.

액터의 행동 관점이란 액터가 유스케이스 시나리오를 주도적으로 수행하도록 하는 것이다. 액터가 유스케이스 시나리오를 주도적으로 수행하기 위해서는 우선, 액터가 행동할 것을 전부 추출하여 시나리오 흐름이 되도록 정의한다. 액터가 행동하면 시스템이 받아서 처리하게 되므로 액터의 행동만을 정의하여도 시나리오가 흘러갈 수 있어야 한다.

시스템 처리 관점이란 요구사항 목표를 달성하기 위하여 시스템이 무엇을 할 것인가를 정의하는 것이다. 시스템이 처리할 것이라고 해서 시스템 처리 방법을 기술적으로 작성하는 것이 아니다. 사용자가 요구사항이 유스케이스 반영된 것을 확인하기 위해서는 유스케이스를 읽고 이해할 수 있어야 하는데, 기술적으로 요구사항을 작성하면 사용자가 이해할 수 없다.

시나리오 관점이란 요구사항 목표를 달성하기 위한 시나리오 흐름을 정의하는 것이다. 고객이 고객정보를 입력하게 되면 시스템이 저장 처리를 수행하는 것이지, 시스템이 저장 처리를 수행한 후 고객이 고객정보를 입력하는 것은 아니다. 이와 같이 액터와 시스템은 시나리오 흐름에 따라

행동해야 하며, 무작위로 행동하는 것이 아니라 정해진 순서에 따라서 행동해야 한다.

액터가 행동하게 되면 시스템이 무조건 액터의 행동을 받아서 처리하는 것은 아니다. 액터가 이벤트를 발생시켜야 비로소 시스템이 처리를 시작하게 된다. 액터가 이벤트를 발생시키지 않으면 시스템이 처리를 시작하게 되는 원인이 없으므로 시스템 처리를 시작할 수 없다. 또한, 시스템 처리가 완료되면 액터가 행동할 수 있도록 이벤트를 제공해야 한다. 이벤트를 제공하지 않았는데 액터가 행동을 한다는 것 또한 근거 없는 행동이 되는 것이다. 액터와 시스템이 근거를 갖고 행동하도록 시나리오 흐름을 정의하는 것이 시나리오 관점이다.

2.1.4 시스템 모델링

시스템 모델링은 사용자 관점에서 작성된 유스 케이스 시나리오를 기준으로 시스템 처리 중심으로 요구사항을 분석하여 시스템 처리 시나리오를 정의하는 단계이다. 요구사항은 사용자가 할 것과 시스템이 할 것을 포함해야 하며 시스템 처리를 통해서 요구사항 목표를 달성한다는 것이 전제조건이다. 즉, 요구사항은 사용자의 행동과 시스템 처리로 구분되며, 사용자의 행동과 시스템 처리 중 어느 것 하나 소홀히 하면 완전하게 요구사항을 정의할 수 없다.

사용자 관점에서 요구사항을 정의하면 사용자의 행동 중심으로 요구사항을 정의하게 되므로 시스템 처리는 피동적으로 정의된다. 이것은 일방적인 각도에서 요구사항을 정의하는 것이므로 시스템이 주체가 되어 행동하고 이것을 사용자가 받아서 행동하는 관점도 필요하다. 그래야 일방적인 각도에서 벗어나 완전하게 요구사항을 정의할 수 있다. 시스템이 주체가 되어 행동하기 위해서는 시스템 처리 중심으로 요구사항을 정의해야 하는데, 이것이 시스템 처리 관점이다.

시스템 처리 관점은 정보기술과 연결된다. 시스

템 목표를 달성하기 위하여 필요한 정보기술에 대해서는 시스템 개발 시작 전에 결정되어 있을 것이지만, 각각의 요구사항에 대하여 어떻게 정보기술을 적용할 것인가는 시스템 처리 관점에서 요구사항을 분석하여 결정해야 한다. 여기서 무엇(What)도 중요하지만 어떻게(How)도 중요하다. 정보기술을 접목하여 효율적이고 합리적으로 요구사항 목표를 달성할 수 있도록 요구사항을 정의하는 것이 요구사항 분석이다.

시스템 처리 중심으로 요구사항을 정의하는 단계가 없다는 것은 요구사항 목표를 달성하기 위한 정보기술의 접목을 분석하는 단계가 없다는 것과 같다. 이것은 완전하게 정보기술을 적용하지 못하는 결과를 초래할 수 있다. 정보기술을 접목하여 요구사항을 매우 효율적으로 처리했던 경험을 분석가는 가지고 있다. 정보기술 접목은 어렵게 느껴지던 요구사항을 쉽게 해결할 수 있으며, 더욱 효과적으로 요구사항 목표를 달성할 수 있다. 이렇게 시스템 처리 관점에서 중점적으로 요구사항을 정의할 수 있는 단계가 필요하다.

완전하게 요구사항을 정의하지 않으면 요구분석 이후 단계에서 사용자와 요구사항을 협의하고 정의해야 한다. 그런데 이때 다시 요구분석 단계부터 수행할 수 없으므로 요구분석 절차에 따라 요구사항을 정의하지 않고 즉흥적으로 요구사항을 정의하게 된다. 그래서 시스템 개발이 지연되고 시스템이 흔들리는 것이다. 요구사항 정의는 요구분석 단계에서 완료해야 한다.

시스템 처리는 데이터 처리를 동반하게 되므로 시스템 모델링 단계에서 데이터 모델링을 병행하여 수행한다. 데이터 모델링은 엔티티(Entity)를 정의하는 개념 데이터 모델링을 의미한다. 시스템 처리에서 발생한 데이터 처리는 데이터 모델에 반영되어야 하고, 데이터 모델링에서 발생한 데이터 처리는 시스템 모델에 반영되어야 한다. 즉, 시스템 모델링과 데이터 모델링은 상호 보완관계를 갖고 데이터 처리를 정의하게 된다.

2.1.5 프로토타입 모델링

프로토타입 모델링은 사용자가 확신을 갖고 요구사항을 확정하도록 하기 위하여 개념적, 추상적으로 작성한 요구사항을 형상화, 구체화하는 단계이다. 즉, 프로토타입(Prototype)을 작성하고 작성한 프로토타입을 사용자와 검토를 반복하여 최종적으로 요구사항을 확정하는 단계이다.

일반적으로 시스템 개념을 파악하기 위하여 화면 중심으로 개략적으로 프로토타입을 작성한다. 그러나, 이벤트 프로세스 모델링은 유저 인터페이스(User Interface)를 포함하여 완전한 형태로 작성하고 디자이너까지 참여하여 디자인까지 한다. 화면/보고서/메뉴를 시스템 설계 단계에서 다시 작성하지 않는다. 실제로 시스템에서 사용하게 될 형태로 프로토타입을 작성한다.

비즈니스 현행 모델링을 통하여 현행 비즈니스 프로세스를 조사하고 문제점을 추출하였으며, 비즈니스 목표 모델링을 통하여 문제점에 대한 해결 방안과 요구사항을 비즈니스 시나리오로 정의하였다. 유스케이스 모델링에서 사용자 관점에서 요구사항을 정의하였으며, 시스템 모델링에서 시스템 처리 관점에서 실질적인 시스템 모습을 만들기 위한 준비를 하였다. 시스템 처리 시나리오를 정의하였으며, 데이터 모델을 작성하였고, 화면과 보고서에 포함할 항목도 정의하였으므로 실제로 사용할 수 있는 프로토타입을 작성할 수 있다. 완전한 형태의 프로토타입은 사용자가 확신을 갖고 향후 시스템의 모습을 확정하기 위한 것이며, 요구사항 변경을 사전에 예방하기 위한 것이다.

요구사항을 확정하기 위해서는 확정에 대한 기준이 있어야 한다. 요구사항을 확정하는 기준은 사용자 측면과 분석가 측면으로 나눌 수 있다. 사용자 측면에서 요구사항 확정은 프로토타입 검토가 더 이상 필요하지 않는 것이다. 그야말로 이젠 바꿀 것도 없고 추가할 것도 없는 상태를 의미한다. 시스템 설계는 요구분석 단계에서 작성한 것을 기준으로 설계자가 수행한다. 시스템 설계를

하면서 사용자와 대화를 하겠지만, 대화 내용이 명확하지 않은 요구사항을 정의하기 위한 것이라면 요구사항을 확정된 것이 아니다. 요구사항에 대한 사용자의 부가적인 설명 없이 설계자가 설계를 할 수 있도록 요구사항을 명확하게 정의한 상태가 분석가 측면에서 요구사항을 확정된 것이다.

요구사항을 확정하더라도 요구사항 변경이 발생하기도 한다. 이것은 프로토타입을 면밀하게 작성하지 않고, 검토를 철저하게 하지 않았기 때문이다. 또한, 변경의 범위가 항목 1~2개를 변경하는 미미한 정도이므로 시스템 개발에 그다지 영향을 미치지 않는다. 상황에 따라 다르겠지만, 이 정도는 요구사항 변경이라고 할 수 없는 면도 있다.

요구사항을 확정하였다고 하여 요구사항을 변경할 수 없는 것은 아니다. 하지만, 요구사항을 확정 후 요구사항을 변경하면 요구사항 변경관리의 통제를 받아야 하며, 요구사항 변경은 사용자의 귀책 사유가 되므로 개발자는 응분의 보상을 받을 수 있다. 하지만, 근본적으로 개발자는 요구사항 변경을 싫어한다. 가다가 되돌아와서 다시 시작하는 것은 개발자를 힘들게 만들며 의욕을 떨어뜨린다. 처음부터 고생을 하더라도 하나씩 마무리 해나가는 맛이 있어야 개발에도 흥이 나며 정해진 기간 안에 시스템 개발을 완료할 수 있다.

2.1.6 요구분석 완료 기준

목표는 가시적으로 측정할 수 있는 기준을 가지고 있으며, 이 기준으로 목표 달성 여부를 판단한다. 요구사항을 정의하고 분석하는 것이 일반적으로 알려진 요구분석의 목표이다. 즉, 요구사항 정의와 분석을 완료하면 요구분석 목표를 달성한 것이다. 그런데, 요구사항 정의와 분석 완료의 의미가 너무 추상적이어서 이것을 기준으로 요구분석 목표 달성 여부를 가시적으로 측정할 수 없다. 요구사항 정의와 분석 완료가 요구분석 완료를 측정하는 기준이 될 수 없다. 지금까지 요구분석 완료에 대한 기준도 없이 요구분석을 수행하고 주관적

으로 판단하여 요구분석을 완료했다고 하였다.

요구분석 완료에 대한 정확한 기준이 없으므로 상세하게 수행한 것도, 개략적으로 수행한 것도 요구분석을 수행한 것이다. 더욱 큰 문제는 요구분석 단계 이후에 부족한 사항이 도출된다는 것이다. 잣대를 대고 엄밀하게 요구분석 완료 여부를 판단하지 않고 요구분석을 완료하였기 때문에 요구사항이 추가되고 변경되는 것이다. 다른 측면에서 생각해보면 요구사항 추가/변경이 발생한 것이 아니다. 완료 기준 없이 요구분석 완료 여부를 판단하였으므로 요구분석을 완료한 것이 아니라, 요구사항 추가/변경 발생이라고 생각하는 시점까지도 요구분석이 진행 중인 것이다.

문서에 작성한 요구사항을 형상화, 구체화한 것이 프로토타입이다. 작성한 프로토타입을 사용자가 더 이상 변경/추가할 것이 없다고 할 때까지 검토와 수정을 반복하여 프로토타입을 확정한다. 프로토타입을 확정한다는 것은 요구사항을 확정된 것이며, 요구분석을 완료한 것이다.

2.2 이벤트와 프로세스

마라톤을 하기 위해 출발선에 서있는 건각들은 출발 신호를 듣고 골인 라인을 향해 달려나간다. 이때 건각들이 달려나가게 하는 원인을 제공한 출발 신호가 이벤트이다. 건각들이 골인 라인을 향해 한발 한발씩 뛰어나가고 있다. 이때 한발씩 뛰는 것이 프로세스이다. 또한, 출발 신호가 나야 달리기 시작하듯이 프로세스는 임의로 시작할 수 없다.

2.2.1 프로세스

요구사항은 다수의 프로세스가 일련의 흐름 형태로 수행되어야 하는 프로세스의 집합이다. 일련의 흐름이란 요구사항을 정의한 순서대로 수행하는 것만을 의미하는 것이 아니다. 조건에 의해 분기를 하거나 예외가 발생하여 예외 처리를 하는 것도 일련의 흐름이다.

인터넷 쇼핑몰에서 `고객이 고객정보를 입력한

다`는 것은 고객이 구매를 하기 위한 비즈니스 시나리오에 속하는 하나의 비즈니스 프로세스이며 비즈니스 모델링에서 사용한다. `고객이 고객정보를 입력한다`는 비즈니스 프로세스는 유스케이스 모델링에서 액터와 시스템이 행동하는 일련의 시나리오로 정의하게 되는데, 이때 액터와 시스템이 행동하는 각각의 행동이 프로세스이다.

2.2.2 이벤트

- 영업사원이 매출전표를 입력한다.
- 영업사원이 <저장> 버튼을 선택한다.
- 시스템은 입력한 매출전표를 저장한다.

상기 시나리오에서 영업사원이 <저장> 버튼을 선택하면 시스템은 입력한 매출전표를 저장하는 프로세스를 수행하게 된다. 만약, 영업사원이 <저장> 버튼을 선택하지 않으면 시스템은 저장 처리를 수행할 수 없다. 즉, 영업사원이 <저장> 버튼을 선택하는 것은 시스템이 저장 처리를 수행하게 되는 원인을 제공한 것이다. 이와 같이 다른 프로세스가 수행하게 되는 원인을 제공하는 프로세스를 이벤트라고 한다.

■ 모델 시나리오

- 1 사은품 담당자가 DM을 출력한다. : [BP1]

■ 이벤트 처리

[BP1]

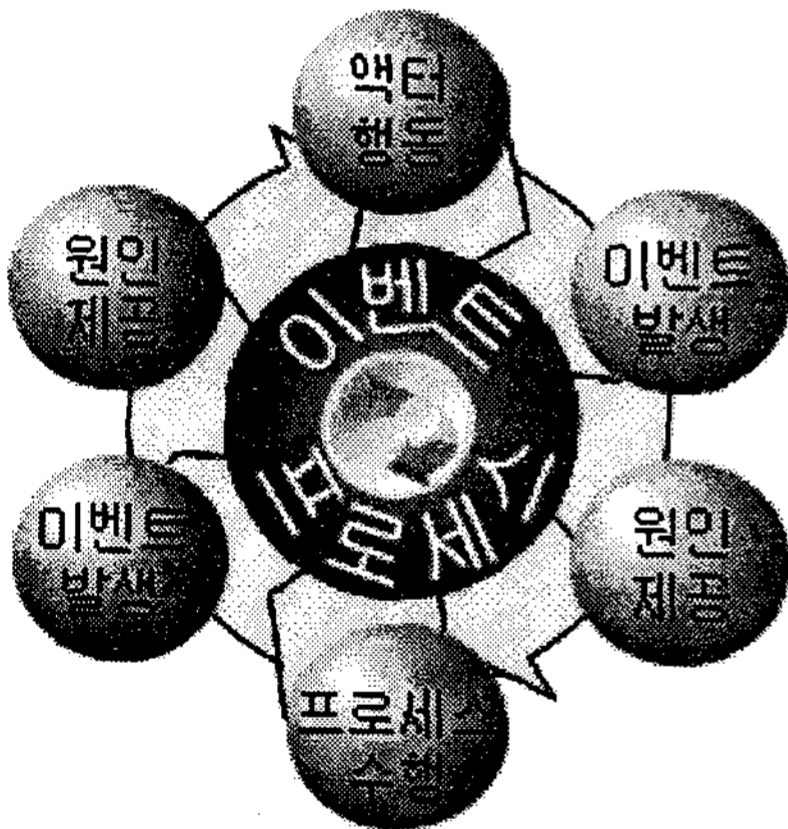
- 1 사은품 담당자가 DM을 발송한다.

모델 시나리오 1번 라인 비즈니스 프로세스는 이벤트 처리 1번 라인 비즈니스 프로세스가 수행하게 되는 동기를 제공한 것이므로 이벤트가 된다. 사은품 담당자가 DM을 발송하기 위해 DM을 출력하게 된다고 생각할 수도 있다. 하지만, 비즈니스 프로세스 흐름을 볼 때 DM을 출력해야 DM을 발송할 수 있으므로 DM을 출력하는 것을 먼저 수행해야 한다. 그러므로 DM을 출력하는 것이 이

벤트가 되고 DM을 발송하는 것이 프로세스가 된다.

2.2.3 시나리오

사용자는 사전에 약속된 시나리오에 의해 시스템을 사용하게 된다. 사용자가 시스템을 사용한다는 것은 화면은 어떻게 하면 표시되고, 표시된 화면에서 상품을 선택하기 위해서는 어떻게 해야 하는가에 대하여 사전에 약속을 한 것이다. 인터넷 쇼핑몰 고객과 시스템 사용 방법에 대한 약속은 하지 않았지만, 사회 통념이 약속을 해준 것이다. 요구사항을 정의하고 분석하는 것은 요구사항 목표를 달성하기 위해 액터와 시스템이 행동할 것을 약속하는 것이고, 약속을 한다는 것은 시나리오를 정의하는 것이다. 시나리오를 잘못 정의하는 것은 사용자와 약속을 잘못하는 것이 된다.



[그림4] 시나리오 흐름

3. 결론

요구사항을 정의하고 분석하는 것이 요구분석의 기본적인 목표는 될 수 있지만, 최종적인 목표가 될 수는 없다. 시스템 개발을 지연시키는 가장 큰 요인은 요구사항이 변경되는 것이다. 홍수가 나기 전에 댐을 만들고 제방을 쌓아 홍수를 예방해야 하듯이, 요구사항 변경이 발생하기 전에 예방해야 한다. 요구사항 정의와 분석을 명확하게 해야 하는 것은 물론이고, 요구사항 변경을 예방하기 위한 전략을 가지고 요구분석을 수행해야 한다. 처

음부터 요구사항 변경이 발생할 것을 염두에 두고, 요구사항 변경을 예방하기 위한 방법으로 요구분석을 수행해야 한다.

이것은 지금까지의 요구분석에 대한 개념을 바꾼 것이며, 한 차원 높게 접근한 것이다. 요구분석에 국한하여 접근한 것이 아니라 시스템 개발 전체적인 시각에서 접근한 것이다.

비즈니스 현행 모델링을 통하여 현행 비즈니스 프로세스를 이해하고 문제점을 추출한다. 비즈니스 목표 모델링을 통하여 문제점에 대한 해결방안과 요구사항을 비즈니스 프로세스 단위로 비즈니스 시나리오를 정의하고 유스케이스로 변환한다. 유스케이스 모델링에서 사용자 관점에서 요구사항을 정의하고, 시스템 모델링에서 시스템 처리 관점에서 요구사항을 정의한다. 프로토타입 모델링에서 사용자가 확신을 갖고 요구사항을 확정할 수 있도록 프로토타입을 작성한다. 이러한 일련의 단계는 요구사항을 점증적으로 명확하게 정의하기 위한 것이며, 요구사항을 확정하기 위한 것이고, 요구사항 변경을 사전에 예방하기 위한 것이다.

요구사항은 근거와 논리를 갖춘 형태로 정의해야 한다. 그 중심에 이벤트와 프로세스가 있다. 이벤트를 발생시켜 프로세스가 수행하게 되는 원인을 제공하고, 이벤트를 받아서 처리한 프로세스 처리 결과가 다시 이벤트를 발생시켜 다른 프로세스가 수행하게 되는 원인을 제공하는 형태가 되어야 한다. 이와 같이 이벤트와 프로세스가 서로 유기적으로 연결된 시나리오 흐름이 되어야 근거와 논리를 갖춘 형태로 요구사항을 정의한 것이다.

[참고문헌]

- [1] 김영보, 요구분석을 위한 Event Process 모델링, 가메 출판사, 2005