

방대성과 복잡성에 대응하는 요구사항분석방법

임상원⁰, 이택, 김도훈, 인호
 고려대학교 컴퓨터학과
 {ckato⁰, comtaek, karmy01, hoh_in}@korea.ac.kr

Requirements Analysis Method for Complexity System

Sang Won Lim⁰, Taek Lee, Do Hoon Kim, Hoh Peter In
 Department of Computer Science & Compute Engineering, Korea University,

요 약

본 논문은 현대사회의 방대성과 복잡성에 대응하여 *WinWin Spiral Model* [Boehm, 1988][Boehm, 1996]을 개선 시켰다. 개선 방법으로는 요구사항을 객체지향적으로 상관관계 표현하는 방법과 어떤 요구사항에 대하여 관련된 객체를 함께 토론하고 조정하는 시간을 가지는 것이다. 또한 요구사항을 구체화 하는데 있어 함께 동의하는 수준까지만 구체화 하여 모든 이해당사자들이 만족하는 안에서 만들어질 시스템의 유연성을 확보하였다.

1. 서 론

요구사항이 방대하고 복잡해짐에 따라 이해당사자들은 요구 사항을 빠뜨리고 넘어 갈 위험이 많아 졌으며, 요구 사항이 복잡해 짐에 따라 충돌을 감지하고, 해결하는 것이 어려워 졌다.

전체적인 요구사항이 서로의 연관관계에 따라 정리되지 않으면 잠재적 요구사항의 충돌을 감지하기 힘들고, 감지된 요구사항의 충돌도 절충 시작점을 찾기가 어렵다.

전체적인 요구사항의 연관관계를 정리하기 위하여 이 논문에서는 요구사항의 깊이란 개념을 사용할 것이다. 요구사항의 깊이란 이해당사자들의 요구사항의 상대적인 구체화 정도이다. 다음과 같은 상황을 보면 쉽게 이해가 될 것이다. 어떤 시스템의 출력 방식에 대한 논의를 하는데 한 이해당사자는 시각적 출력을 요구하고, 다른 한 이해 당사자는 명암 비 300:1을 요구 한다고 하자. 이런 경우 요구사항의 깊이가 다르다고 한다.

이 두 가지 요구사항은 충돌여부를 아직 알 수 없다. 시각적 출력의 요구사항은 명암 비 요구사항을 포함할 수도 있고 안 할 수도 있다. 여기서 시각적 출력을 요구한 이해 당사자가 그의 요구사항을 모두 말하지 않은 것이라면 잠재적으로 이의를 제기할 위험을 가지고 있다.

이 둘의 요구사항 깊이가 서로 다르기 때문이다. 이러한 문제점을 해결 하기 위해 *WinWin Spiral Model* [Boehm, 1996]을 기반으로 개선된 모델을 제안 한다..

2. 관련연구

2.1 WinWin Spiral Model

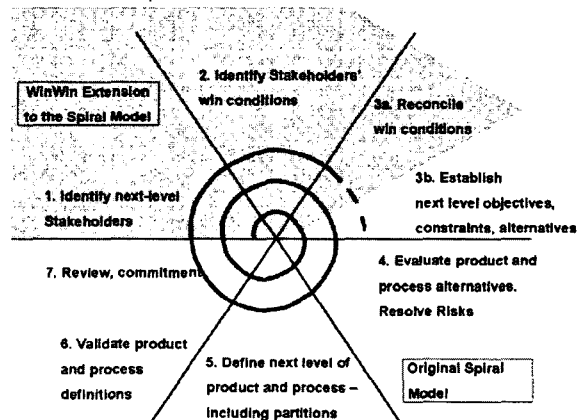


그림 1: WinWin Spiral Model [Boehm, 1996]

위 그림1은 WinWin Spiral Model이다. 이 모델은 그림과

같이 다음과 같은 일련의 과정을 거쳐 요구사항을 분석한다.

어두운 부분이 원래의 Spiral 모델에서 WinWin으로 확장된 부분으로 이 부분에서 이해당사자들의 요구사항이 절충된다. 그 프로세스는 다음과 같다.

1. 이해당사자들을 식별한다.
2. 각 이해당사자들이 Win Condition 을 제시한다.
3. 충돌되는 Win Condition 절충한다.

Win condition은 이해당사자들의 시스템에 관한 목표이다. 그것들이 상충 될 때에 Step3에서 그것을 절충한다. 그 절충 방법은 WinWin Negotiation으로 아래와 같다.

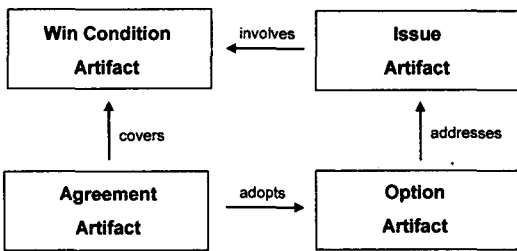


그림2: WinWin Negotiation Model[1]

Win Condition은 이해당사자의 새로운 시스템에 관한 목표를 담고 있다. 만약 Win Condition이 충돌이 일어나지 않으면 그것은 동의 되는 것으로 간주된다. 반대로 충돌이 일어나면 이해당사자들은 그 이슈와 관련된 대안을 제시할 수 있는 옵션이 허락된다. 그리고 그 옵션이 채택되면 합의가 이끌어 지게 된다. [2]

이 모델에서는 그림 1의 프로세스가 충분한 의사를 주고 받을 만큼 반복된다는 것을 가정한다. [2]

3. 개선된 모델 제안

3.1 모델 제안

그림 3은 필자가 정의한 Requirements Tree 이다. Requirements Tree란 미리 주어진 기본적인 구성요소에 대하여 함께 생각하여 결정한 사항을 시각화한 것이다. (WinWin Spiral Model [Boehm, 1996] 에서 기본적인 구성요소는 주어진다고 가정한다. [3]) S가 미리 주어진 구성요소라고 하면 1,2는 그것들의 요구사항 가능성 들이다. 이해당사자들은 WinWin Negotiation방법으로 여러 요구사항 가능성 중에서 하나 또는 복수개의 경우를 선택하고 그것을 더 구체화 시킬 것인가를 판단한다.

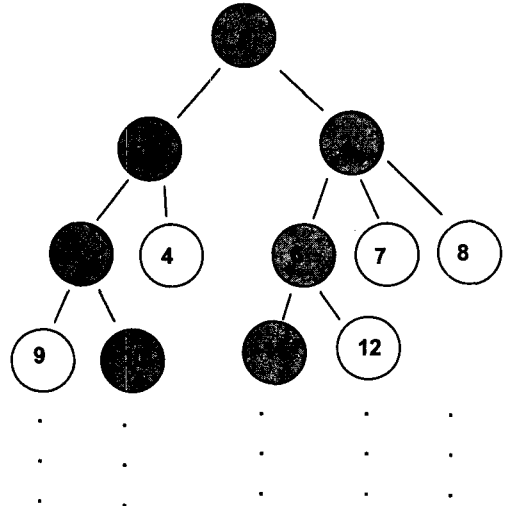


그림3: Requirements Tree

이해당사자들이 더 구체화된 요구 사항이 필요하다고 동의하면 구체화된 다음 단계의 요구사항들을 함께 제시하고 앞의 과정을 반복한다. 이 과정은 이해당사자들이 만족하는 상태까지 반복된다 Theory W[4].

구체화 과정을 반복 하기 전 앞서 구체화 시킨 요구 사항은 Degree Freeze 하여 선택된 사항은 외부적인 변화나 다른 요소와의 충돌이 없을 경우 구체화 시키는 행위를 제외하고 변경을 금지한다.

이해당사자들이 함께 요구사항을 구체화 시키는 방법을 정리하면 다음과 같다.

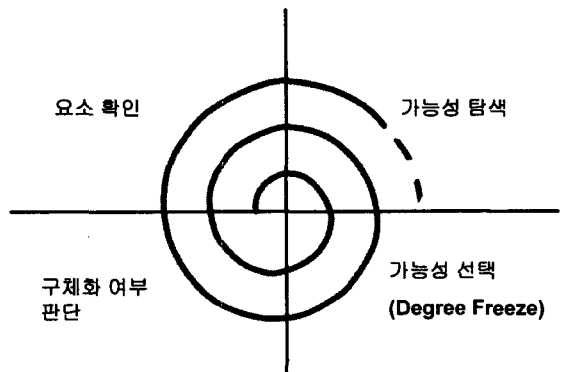


그림4: Requirement Tree 구체화 모델

1. 요소확인
2. 가능성 탐색
3. 가능성 선택 (Degree Freeze)
4. 구체화 여부 판단(반복)

위 과정에서 그려진 *Requirements Tree*는 문서화 하여 차 후에 충돌이 발생하였을 때 어느 단계에서 절충이 필요한지를 판단할 때 사용 한다. 또한 절충되어 변화된 요구사항 아래에 있던 자식 요구사항들은 자동으로 폐기 된다.

이 프로세스는 *WinWin Spiral Model [Boehm, 1996]* Step2,3 에서 진행되어 이해당사자들이 요구사항을 생각하고 절충하는데 도움을 준다..

요구사항 충돌 발생시 *WinWin Negotiation Model[1]* 의 프로세스를 이용 절충한다.

절충은 모두가 만족하는 상태까지 충분한 시간을 두고 이루어 진다 Theory W[4].

3.2 적용

그림2에서 S를 컴퓨터 시스템의 출력부분이라고 했을 때 대안1은 청각적 방법, 대안2는 시각적 방법으로 가능성이 조사 되었다. 이해당사자들은 이 두 가지 모두가 필요하다고 동의하고 두 가지를 모두 선택 하였다. 그리고 더 구체화가 필요하다고 동의하고 먼저 청각적인 방법의 구체화에 들어 갔다. 청각적인 출력방법으로 스피커(대안3)와 이어폰(대안4)가 탐색되었고 그 중 스피커(대안3)이 선택되었다. 이해당사자들은 더 구체화가 필요하다고 동의하고 2채널스피커(대안9)와 5.1채널스피커(대안10)을 고려하여 5.1채널 스피커(대안10)을 선택하였다. 이해당사자들은 더 이상의 구체화는 필요 없다고 동의하고 여기서 청각적인 출력 방법에 대한 구체화를 멈추었다. 시각적인 출력 방법도 이와 유사하게 진행 되었다.

3.3 효과

첫째, *Requirement Tree*를 그리고 그것을 구체화 하는 모델을 사용 하였을 때 이해당사자들은 쉽게 같은 수준에서 구체화 된 요구사항을 논의 함으로써 서로의 의사전달이 보다 효과적이다.

둘째, 이해당사자들이 요구사항 가능성에 대하여 함께 탐색 하기 때문에 요구사항을 빠뜨리지 않고 논의 할 수 있게 되었다. 혹시나 있을 수 있는 요구사항 누락의 위험을 줄였다.

셋째, 충돌이 일어났을 때 단계별로 절충하면서 기존의 요구사항을 기반으로 최소한의 변화로 신속히 대처할 수 있고, 변화가 생긴 요구사항에 관련된 하위 요구사항을 신속히 제거 할 수 있다.

넷째, 요구사항의 구체화 단계를 서로가 만족하는 수준까지로 한정함으로 불필요한 제약 사항이 줄고, 전체 요구사항 분석 시간이 단축 되었다.

4. 결론

본 논문에서는 *WinWin Spiral Model [Boehm, 1996]* 모델에서 Step2,3과정을 개선하였다. 이해당사자들은 함께 요구사항의 가능성을 탐색하고 단계별로 구체화 해가는 과정에서 요구사항누락 위험을 줄였으며, 서로가 만족하는 범위 안에서 요구사항을 함께 구체화 시켜 불필요한 제약사항을 줄이고 시간을 절약 할 수 있게 되었다. 이로, 이해당사자들은 적절한 시간 안에 서로가 원하는 요구조건을 만족할 수 있다 Theory W[4].

참고 문헌

- [1] B. Boehm and H. In, "Aids for Identifying Conflicts Among Quality Requirements," One of best papers at International Conference on Requirements Engineering (ICRE96), Colorado, April 1996, and IEEE Software, March 1996
- [2] Egyed, A., Boehm, B. "Comparing Software System Requirements Negotiation Patterns", to appear in Journal for Systems Engineering, 1999.
- [3] Egyed, A. and Boehm, B., "Analysis of System Requirements Negotiation Behavior Patterns," Proceedings of the International Conferences of Systems Engineering (INCOSE), July 1997.
- [4] Boehm, B. and Ross, R., " Theory W Software Project Management: Principles and Examples," IEEE Trans. on Software Engr., July 1989, pp. 902-916.