

웹기반 소프트웨어 요구 조정 시스템의 설계 및 구현

권 기 태†

요 약

소프트웨어 개발 과정에서 모든 stakeholder의 효율적인 협력과 소프트웨어 요구조정은 성공적인 소프트웨어 개발을 위한 중요한 전제 조건 중의 하나이다. 지리적으로 분산된 stakeholder들이 네트워크로 연결된 컴퓨터를 통하여 서로의 정보를 공유하며 요구 조정 작업을 수행하도록 지원하는 컴퓨터 이용 소프트웨어 요구조정 시스템의 기본적인 목표는 모든 win 조건을 통합하고 win 조건의 충돌을 해결하여 일치시키기 위한 작업이다. 그러나 기존의 요구 조정 시스템은 전용 시스템 도입을 필요로 하여 시스템이 설치된 장소에서만 요구 조정 작업이 가능하였다. 따라서 특정 플랫폼과 네트워크 환경에 종속되는 약점을 가지고 있었다. 본 논문에서는 이러한 기존의 요구 조정 시스템의 문제점을 해결하여 이질적인 웹 브라우저, 인터넷 연결, 운영 체제, 플랫폼 아래의 다양한 stakeholder들이 WWW을 이용하여 전용 시스템의 지원이 없이도 소프트웨어 요구 조정 작업을 수행할 수 있는 웹 기반 소프트웨어 요구 조정 시스템을 제안한다. 제안한 시스템은 WinWin 나선형 프로세스에 의해 구동되며, 중앙 실행 방식과 복제 실행 방식을 통합한 복합 실행 방식의 구조를 가진다. 제안한 시스템을 다양한 형태의 인터넷 연결과 운영 체제를 가진 stakeholder들이 웹 브라우저만을 이용하여 요구 조정 작업이 가능한지의 여부를 확인하였다.

Design and Implementation of Web-based Software Requirements Negotiation System

Ki-Tae Kwon†

ABSTRACT

One of the most important prerequisites for a successful software process is the collaboration and software requirements negotiation of all stakeholders in the software development process. Remote users using computer networks can negotiate software requirements by computer supported system, and can share their informations. The goal of software requirements negotiation system is an integration of all win conditions and an agreement after resolution of conflicts. The existing systems need an exclusive system and must be dependent on specific platform and network. Users must have the knowledge of all stakeholder's status and use homogeneous collaborating applications. This paper presents the Web-based software requirements negotiation system for the purpose of resolution of the existing systems' problems. The Web-based software requirements negotiation system can be driven by WinWin Spiral model, and it is based on hybrid execution method. The proposed system is validated and tested on heterogeneous environments.

※ 본 연구는 한국과학재단 98년도 핵심연구과제(과제번호 : 981-0923-122-2)의 지원을 받았다.

† 종신회원 : 강릉대학교 컴퓨터학과 교수

논문접수 : 1999년 10월 11일, 심사완료 : 1999년 11월 13일

1. 서 론

최근 소프트웨어 공학 국제 학술회의의 화두(話頭)는 소프트웨어 프로젝트의 가장 중요한 성공 요소가 소프트웨어 요구(requirements)의 조정 기법(negotiation techniques)이라는 것이었다[1]. Tom DeMarco는 “요구가 어떻게 조정되는가 하는 것이 요구가 어떻게 명세화 되는가 하는 것보다 더욱 중요하다”라고 언급했다[2]. Ed Yourdon은 “조정(Negotiation)이 프로젝트를 실패로부터 피할 수 있는 최선의 방법이다”라고 말했다[3]. 또한 Mark Weiser는 “합의에 도달한 문제는 도구, 프로세스 성숙도(maturity), 설계 기법 등과 같은 요인보다 프로젝트의 성공에 더욱 중요하다”라고 결론을 맺었다[4].

소프트웨어 개발은 주로 수백 명까지 도달하는 그룹의 작업으로 개발자가 소비하는 시간의 약 절반이상이 다른 사람과의 상호작용에 소비된다고 한다. 소프트웨어 개발 과정에서 stakeholder는 대개 개발자, 고객, 사용자, 유지보수 담당자, 그리고 그 밖에 프로젝트에 관여하는 사람들이다[5]. 소프트웨어 개발에 참여하는 stakeholder간에 모호하지 않고 효과적으로 상호작용하는 것은 성공적인 제품을 개발하는데 있어 중요한 단계가 될 수 있다. 그러나 stakeholder가 문제를 제대로 해결하지 못하면 상호작용 그 자체가 아무리 효율적이라 하더라도 충분하지 못할 수 있다. W 이론[6]은 모든 stakeholder를 winner로 만드는 경우에만 프로젝트가 성공하여 바른 제품이 만들어지라는 것을 보장한다. 이를 위해서는 중요한 모든 stakeholder가 개발 과정에 참여할 수 있도록 보장해야 한다. 또한 stakeholder의 효율적인 협력은 성공적인 제품을 만들기 위한 목표를 달성하는데 있어서 중요한 전제 조건이다.

오늘날 급속도로 발전하는 기술과 경쟁이 치열한 시장을 대상으로 하는 복잡한 소프트웨어의 개발은 stakeholder가 개발 초기 단계부터 참여해야 하므로 의사결정을 지원하는 협력 모델과 시스템이 요구된다. 이를 위한 요구 결정의 동적 모델은 개개의 stakeholder의 목적과 상호간의 작용의 모델링, 소프트웨어 요구 도출 및 요구 결정의 모델링, 목적 및 원리 등의 모델링을 포함한다.

요구 분석은 소프트웨어 요구를 획득하고 표현하기 위한 체계적인 틀을 제공할 뿐만 아니라 소프트웨어 공학의 중요 부분을 구성한다. 협력적인 소프트웨어

개발 환경에서 모든 stakeholder의 관점은 통합되어야 하고 조정되어야 한다. 오직 사용자 요구에만 초점을 맞춘 과거의 요구 공학은 사용자의 관점이 아닌 다른 stakeholder의 관점에 맞지 않으므로 실패하였다. 또한, 사용자 요구는 종종 잘못 해석되기도 하는데, 그 이유는 다른 주요 stakeholder가 요구 명세화 단계에 포함되지 않기 때문이다. 게다가 일부 내용이 요구 명세화 단계에 변경되는 것은 매우 일반적이다. 예를 들어, 예산 삭감은 이전의 합의를 무효로 만드므로 변경을 수용하기 위해 변경 관리(change control)가 필요하다. 개발 조직의 요구 조정을 가능하게 하기 위해, 컴퓨터 지원 협동 작업(CSCW)을 설정하는 그룹웨어는 그룹의 결정을 유도할 뿐만 아니라 지리적, 시간적 간격을 메우는데 필요하다.

소프트웨어의 복잡도가 증가함에 따라 소프트웨어의 개발을 위해 많은 개발 인력이 필요하게 되었고 소프트웨어 개발 환경도 개발자들의 요구 사항을 반영하여 다양한 기능을 지원하게 되었다. 이러한 소프트웨어 개발 환경은 소프트웨어 개발 인력이 모두 한 곳에 모여 개발 프로젝트를 진행한다는 가정 하에 독립적(stand-alone) 방식으로 발전되어 왔다. 즉, 하나의 고성능 워크스테이션에 소프트웨어 개발 환경을 구축하고 프로젝트 구성원들은 모두 여기에 접속하여 개발 작업을 수행하는 형태를 취해 왔지만 네트워크가 고속의 데이터 전송을 지원하고 컴퓨터의 성능이 향상됨에 따라 이러한 독립적 방식의 단점을 개선한 분산 소프트웨어 개발 환경의 유용성이 부각되고 있다. 분산 소프트웨어 개발 환경은 소프트웨어 개발 환경을 네트워크로 연결된 여러 대의 컴퓨터에 분산시켜 개발자에게 보다 편리한 환경을 제공하며 전체적인 성능도 향상시킬 수 있는 이점을 가지고 있다. 또한 컴퓨터 네트워크를 이용하면 지리적으로 멀리 떨어져 있는 컴퓨터에서 분산 개발 환경에 접속하여 대용량의 멀티미디어 데이터를 고속으로 주고받을 수 있다. 즉, 지리적으로 분산된 stakeholder들이 네트워크로 연결된 컴퓨터를 통하여 서로의 정보를 공유하며 요구 조정 작업을 수행하도록 지원하는 컴퓨터 지원 요구 조정 작업의 필요성이 대두되고 있으며 이를 지원하는 그룹웨어(groupware)[7]가 소프트웨어 공학에서의 새로운 분야로 각광받고 있다.

그러나 기존의 소프트웨어 요구 조정 시스템은 전용 시스템 도입을 필요로 하며, 시스템이 설치된 장소에서만 공동작업이 가능하기 때문에 특정 플랫폼과 네트

워크 환경에 종속될 뿐만 아니라 사용자들이 참가자들의 네트워크 접속 상황을 사전에 인지해야 하며, 반드시 동일한 종류의 공동작업 응용 프로그램을 사용해야 했다. 본 논문에서는 이러한 기존의 소프트웨어 요구 조정 시스템의 문제점을 해결하기 위해 이질적인 웹 브라우저, 인터넷 연결, 운영 체제, 플랫폼 아래의 다양한 stakeholder들이 WWW을 이용하여 전용 시스템의 지원이 없이도 소프트웨어 요구 조정 작업을 수행할 수 있는 웹 기반 소프트웨어 요구 조정 시스템을 제안한다. 본 논문의 구성은 다음과 같다. 1장의 서론에 이어서 2장에서는 WinWin 나선형 모델에 대해 소개한다. 3장은 본 연구에서 제안하는 웹 기반 요구 조정 시스템의 장점과 문제점을 분석하고, 4장에서는 웹 기반 소프트웨어 요구 조정 시스템의 설계 및 구현에 대해 기술한다. 마지막으로 5장에서는 결론과 향후 연구 과제에 대해 언급한다.

2. WinWin 나선형 모델

2.1 W 이론

소프트웨어 요구를 실현하는 기존의 소프트웨어 개발 과정과 지원 시스템은 다양한 stakeholder의 관심을 소프트웨어 개발 과정과 산출물(product)에서 순차적으로 반영한다. 사용자 요구들의 많은 모호한 표현이 시스템 분석가에 의해 독단적으로 명세서로 변환되어 종종 중요한 측면에서 부정확하다. 이러한 명세서들은 일반적으로 최소의 사용자가 참여하여 개발자와의 계약을 정의하기 위해 이용된다. 이러한 최초의 산출물은 점증적 개발(incremental development)의 초기 증분(increment)에 불과하고 stakeholder의 상세한 관심은 일반적으로 초기의 증분이 운영될 때까지 탐색되지 않는다. 이러한 점이 stakeholder간의 win 조건을 얻기 위해 산출물을 변환하는 것을 매우 어렵게 한다.

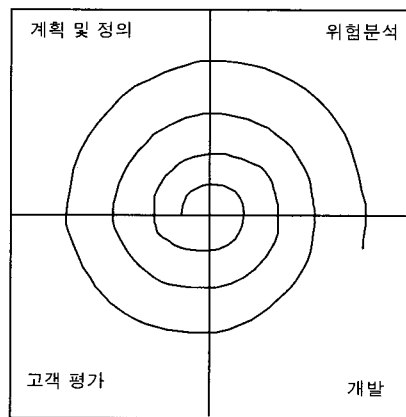
또한 소프트웨어 개발 과정의 각 단계에서의 순차적인 참여는 다양한 stakeholder들 사이의 win-lose 상황을 유도한다. 임시변통의 간략한 제품을 만드는 것은 비용이 적게 들어 소프트웨어 개발자와 고객에게는 win 조건이 되겠지만, 사용자와 유지보수 담당자에게는 lose 조건이 될 것이다. 추가 비용을 부담하는 계약으로 소프트웨어 제품에 특별한 기능을 추가하는 것은 개발자와 사용자에게는 win 조건이지만 고객에게는 lose 조건이다. 경쟁적인 개발자 사이의 가격 경쟁에

의해 결정된 비용은 고객과 사용자는 win 조건이 되나 개발자를 lose 조건으로 유도하게 된다.

실제로 이러한 win-lose 상황에서는 stakeholder 누구도 winner가 될 수 없다. 왜냐하면 임시변통의 간단한 제품은 개발자의 평판을 떨어뜨리고 고객에게 높은 유지보수 비용을 요구하게 된다. 특별한 기능을 추가하는 것은 고객의 예산을 소모하게 하여 제품의 더욱 필수적인 기능을 빠뜨리게 한다. 부정확한 저가의 가격 결정은 품질이 낮은 제품을 만들고 이는 고객의 비용 증가와 원하는 품질을 달성하기 위해 납기를 지연시킨다. 이러한 상황을 해결하기 위해, 소프트웨어 프로세스의 정의 및 개발 단계에서 소프트웨어 제품의 stakeholder의 협력적인 참여를 명시적으로 강조하는 소프트웨어 생명주기 모델을 필요로 한다.

2.2 WinWin 나선형 모델

(그림 1)의 나선형 모델(spiral model)[8]은 고전적 생명주기 모델과 프로토타이핑 모델에서 장점을 따고 이들 패러다임에 없는 새로운 요소인 “위험분석(risk analysis)”을 추가하여 만든 모델이다. 그림에서 각 사이클은 다음의 네 가지 주요 행위를 나타낸다.



(그림 1) 나선형 모델

- 제품과 프로세스의 목적, 제한, 대안을 결정하는 계획
- 목적과 제한에 관한 대안을 분석하고 위험요소의 주요 원인을 인식하고 해결책을 찾는 위험 분석
- 다음 단계의 제품을 개발하고 개선
- 결과를 평가한 후, 다음 사이클을 준비하고 계획을 갱신

나선형 모델에서 나선을 돌 때마다 점진적으로 보다 완전한 소프트웨어 버전들이 만들어진다. 나선을 처음 한 바퀴 도는 동안에 목적, 대안, 제한요소가 정의되며, 위험요소가 알려지고 분석된다. 만일 위험 분석이 요구에 불확실성을 발견하게 되면, 프로토타이핑 모델이 사용되어 stakeholder들을 도와준다. 시뮬레이션이나 그 밖의 모델들이 그 문제를 더 자세히 정의하고 요구를 재정의하는데 이용된다. 고객의 평가에 의해 수정이 제안되고, 고객의 입력이 다음의 계획과 위험 분석의 기본이 된다. 위험이 크면 프로젝트는 끝나게 된다.

새로운 소프트웨어 프로세스 모델은 일반적으로 증명되는데 수년이 걸린다. 나선형 모델은 1978년에 최초로 나왔고, 처음에는 1982년부터 1988년까지 TRW의 내부 프로젝트에 시도되었으며, 1988년부터 대규모 프로젝트로 확대되었고 문서화가 충분히 이루어진 개발 방법이다. 그러나 기존의 나선형 모델은 어디에서 다음 단계의 목표와 제한 사항이 나오고, 이것들이 옳은지의 여부를 결정하는 문제 때문에 모델이 확장되었다. W 이론의 개념은 (그림 2)에 보이는 것처럼 WinWin 나선형 프로세스 모델로 결합되었다[9, 10]. 나선형 모델의 각 사이클은 중요한 모든 stakeholder와 그들의 win 조건을 미리 인식하는 방식으로 W 이론이 결합되어 각 나선 사이클에 다음의 세 가지 영역이 추가되었다.



(그림 2) WinWin 나선형 프로세스 모델

- 시스템이나 서브 시스템의 주요 stakeholder를 인식
- 시스템이나 서브 시스템에 대한 stakeholder의 win 조건을 인식
- stakeholder의 win 조건들을 조정하여 win-win 조건을 유도

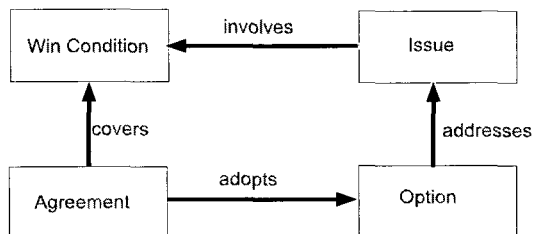
개선된 나선형 모델은 협력 작업의 기초를 제공하며, 소프트웨어 요구를 동시에 분석하고 위험을 해결할 수 있도록 하였다. WinWin 나선형 모델의 두 가지 주요

과정은 stakeholder와 그들의 win 조건을 인식하는 단계와 win-win 관계를 협력적으로 유도하기 위해 win-lose와 lose-lose 상황을 조정하는 단계이다. 프로세스의 주 측면은 제품 품질과 위험요소의 고려를 의사 결정 단계로 도입하고 위험요소와 요구의 충돌을 해결하기 위한 프로세스를 도입했다는 것이다. 프로세스 모델을 지원하기 위해 적절한 양의 통신과 협력이 소프트웨어 개발 과정의 모든 사이클 동안 보장되도록 보증하기 위한 모델과 그룹웨어 도구가 개발되어야 한다.

2.3 WinWin 요구 조정 모델

복잡한 소프트웨어의 개발은 상이한 stakeholder를 포함하므로 이들간의 협력적인 작업을 위한 컴퓨터를 이용한 지원이 필요하다. WinWin 기법은 소프트웨어 생명주기의 요구 분석 단계에서 협력적으로 stakeholder의 도메인 독립적인 해결책을 제공하는 것이다.

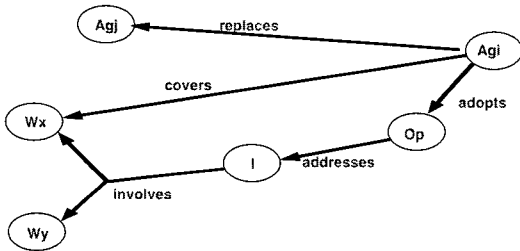
WinWin 요구 조정 모델은 (그림 3)과 같이 Win Condition, Issue, Option, Agreement의 네 가지 개념적 객체를 이용하여 모델링된다. Win Condition은 stakeholder의 목표와 새로운 시스템에서 원하는 조건을 나타낸다. 만일 Win Condition이 논쟁의 여지가 없으면 Agreement로 채택된다. 그렇지 않으면 Issue는 Win Conditions 사이의 위험요소와 불확실성 사이의 충돌을 기록하기 위해 생성된다. Option은 stakeholder가 Issue를 해결할 수 있는 전략을 제시하는 것을 허용한다. 마지막으로 이러한 해결책을 사용하여 Agreements로 유도할 수 있다. 이 객체들 각각은 조정을 위해 도움이 되는 정보를 포함한다.



(그림 3) WinWin 요구 조정 모델

(그림 4)는 WinWin 요구 조정 모델의 객체를 이용한 결정 원리와 이들 사이의 관계를 나타내는 링크 타입의 전형적인 추상 구조를 보여준다. 그림에 보이는 대로, Issue(I)는 involve 관계를 통해 하나 이상의 Win

Condition(W_x 와 W_y)에 관계된다. Issue(I)를 해결하기 위한 Option(Op)은 address 관계를 통해 Issue와 관계된다. Option 선택(Op)에 기반한 Agreement(Ag_i)는 adopt 관계를 통해 Op와 관계된다. Agreement(Ag_i)는 Win Condition에 대해 cover 관계를 가지고, 이전의 합의를 대체할 때에는 replace 관계를 가진다.



(그림 4) WinWin 요구 조정 모델의 결정 원리와 추상 구조

3. 웹 기반 요구 조정 시스템

그룹웨어의 구현을 위해서는 그룹웨어를 사용하는 사용자들이 어떤 형태의 공동 작업을 수행하는가를 파악할 필요가 있다. 공동 작업의 형태에는 구성원이 들어 같은 시간과 같은 장소에서 공동 작업을 수행하는 동기적 모드(synchronous mode), 같은 시간과 다른 장소에서 공동 작업을 수행하는 분산 동기적 모드(distributed synchronous mode), 다른 시간과 같은 장소에서 공동 작업을 수행하는 비동기적 모드(asynchronous mode), 그리고 다른 시간과 다른 장소에서 공동 작업을 수행하는 분산 비동기적 모드(distributed asynchronous mode)의 네 가지 유형이 있다[11]. 본 논문에서 구현되는 웹 기반 소프트웨어 요구 조정 시스템은 동기적 모드와 분산 동기적 모드와 같은 형태의 공동 작업을 지원하는데 중점을 두고 설계하였다. 즉, 지역적으로는 투명하고(transparent) 시간적으로는 동기적으로 수행되는 요구 조정 작업을 지원하도록 하였다.

Stakeholder는 이 시스템을 win 조건, 결과적인 충돌, 가능한 옵션, 그리고 최종적인 해결책을 인식하기 위해 사용하므로, 목표는 stakeholder의 win 조건을 통합하고 충돌을 해결하여 일치시키기 위한 작업이다. 관련 작업은 다중 관점(multiple viewpoints)의 표현에 기초한 요구 분석을 포함한다. 즉, 프로세스에서 다자간(multi-part)의 참여를 시도하기 위해 W 이론을 요

구 생성을 위한 기초로 사용하여 win 조건을 유도하고 win-win 해결책의 완성에 초점을 맞추며 COCOMO [12] 등과 같은 도메인 독립적인 분석 도구의 사용을 지원한다. 웹기반 요구 조정 시스템은 상위 레벨에서 WinWin 나선형 프로세스에 의해 구동된다.

3.1 웹기반 요구 조정 시스템의 장점

WWW는 분산 하이퍼미디어를 기반으로 하는 인터넷 서비스로 네트워크로 접속된 다른 컴퓨터내의 문서나 서비스까지 확대될 수 있다. URL을 이용하면 인터넷에서 제공하는 대부분의 서비스들과 문서들을 표시할 수 있으며, 인터넷의 모든 자원들을 일관된 형식으로 나타낼 수 있다. HTTP 프로토콜은 클라이언트가 서버에 HTML 문서를 요청할 때 사용하는 프로토콜이다. HTML 문서는 하이퍼링크나 이미지를 지정할 수 있는 문서이다. WWW는 이런 다양한 특성과 함께 HTML 문서를 저장하는 서버와 클라이언트 프로그램인 브라우저로 구성된다. WWW상에서의 요구 조정 시스템은 하이퍼미디어 정보 집근의 편리성외에도 여러 장점을 가질 수 있다.

WWW의 클라이언트 프로그램인 브라우저는 인터넷에 연결된 대부분의 플랫폼에 이식되어 있어서 요구 조정 시스템을 특정 전용 시스템 설치 없이도 stakeholder들간에 작업이 가능하게 될 것이다. WWW는 데이터 전송 속도의 차이만 있을 뿐 서비스의 차이가 없으므로 네트워크 접속상황에 크게 영향을 받지 않고 사용이 가능하다. WWW에서는 모든 정보가 서버에 저장되며, 브라우저는 단지 그 내용을 브라우징하는 역할을 담당하게 되므로 브라우징하는 내용에 일관성을 유지할 수 있다. 요구 조정 시스템에서도 서버가 요구 정보를 저장하면 일관성 유지가 용이하고 stakeholder들이 세션과 관련된 모든 정보를 서버에서 전송받아 사용하므로, 정보를 사전에 인지해야 할 필요성이 없어진다. WWW와 요구 조정 시스템을 연동할 경우 정보를 브라우징하면서 요구 조정을 진행할 수 있다.

3.2 웹기반 요구 조정 시스템의 문제

WWW에서 사용되는 HTTP 프로토콜은 브라우저가 서버에 요청할 때만 서버와 접속이 일어나고, 같은 브라우저에서 발생한 각 세션이 독립적인 프로토콜이다. 이 프로토콜은 각 세션의 연계성이 없어서 동일한 서

버를 이용하는 브라우저라도 서로를 인지할 수 있는 방법이 없고 서버와 지속적인 연결을 할 수 없기 때문에 WWW와 요구 조정 시스템을 결합하는 여러 문제를 해결해야 한다. 요구 조정 시스템은 서버와 stakeholder, 그리고 stakeholder들간에 상호작용을 요구하지만, WWW는 브라우저의 HTML 문서 요구에 대한 응답을 서버가 해주는 단순한 프로토콜 구조이므로 동시에 서버에 접속하는 브라우저들 상호간에 인지 방법과 상호작용이 불가능하다. 요구 조정 시스템이 상호작용 기능을 가지기 위해서는 WWW상에서의 상호작용 방안이 제공되어야 한다. HTTP를 이용하여 브라우저가 요구한 각 세션 요청은 서버 측에서 개별적으로 처리되어 관계성이 거의 없고 서버와 브라우저간에 지속적인 세션 연결이 불가능하다. 그러나 요구 조정 시스템에서는 서버와 클라이언트간에 지속적인 연결이 유지되어야 공동작업의 진행이 가능하다. 프로토콜을 수정하면 기존의 서버나 브라우저를 사용할 수 없고, 반대로 서버나 브라우저를 수정하면 요구 조정을 위해 수정된 프로그램을 설치해야 한다. 그러나 WWW 요소를 수정하게 되면 요구 조정전용 시스템의 도입과 별 차이가 없게 되므로 요구 조정 기능을 추가하되 호환성을 유지해야 한다.

4. 웹기반 요구 조정 시스템의 개발

WinWin 나선형 모델을 시험하기 위한 연구가 USC CSE와 관련 기관을 중심으로 진행되고 있다[13]. 우리는 이러한 WinWin 나선형 모델을 지원하는 WinWin 요구 조정 시스템을 WinWin 나선형 모델에 임각하여 개발하였다. 이러한 WinWin 나선형 모델의 응용은 네개의 주요 나선 사이클을 포함한다[1].

- 사이클 0 : WinWin 나선형 모델의 일련의 응용을 위한 타당성 결정
- 사이클 1 : 개개의 응용의 타당성 결정
- 사이클 2 : 각 응용을 위한 적절한 프로젝트 이정표를 기록
- 사이클 3 : 각 응용을 위해 수행 가능한 프로젝트의 초기 운영 가능성을 기록

4.1 Java의 이용

WWW상에서의 요구 조정 시스템 구현을 위해서는

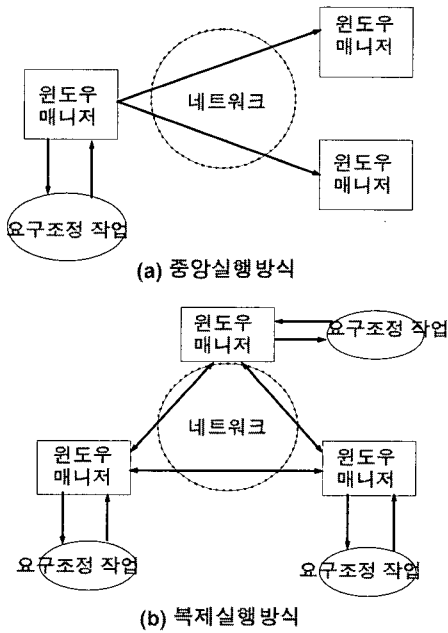
WWW에 요구 조정을 위한 기능을 추가하기 위해 Java를 이용한다. Java의 이동 코드 기술은 실행 가능한 코드가 네트워크를 이동하여 도착한 플랫폼에서 실행되는 기술로 Java는 WWW와 연동 가능한 언어로서 브라우저에서 기본적으로 지원된다. WWW 서버에서 전송된 Java 애플릿은 브라우저에서 독립적인 응용으로서 동작하게 되며, 서버로부터 전송되므로 브라우저에 여러 기능을 동적으로 추가하는데 사용될 수 있다. 이렇게 서버에 설치된 애플릿이 전송되어 사용자의 브라우저에서 실행되므로 stakeholder는 브라우저만으로도 요구 조정을 진행할 수 있어, 전용 시스템 도입 문제를 해결할 수 있다. 즉, WWW와 호환을 유지하면서 새로운 기능을 브라우저에 추가하는데 사용될 수 있다.

Java 언어로 작성된 응용 프로그램은 컴파일시 플랫폼 독립적인 코드가 생성되어 하드웨어나 운영체제의 종류에 상관없이 JVM(Java Virtual Machine)상에서 수행되므로 Java로 응용 프로그램을 작성할 경우 브라우저가 이식된 다른 기종의 플랫폼에서 실행될 수 있어서 수정없이 다른 기종 플랫폼 stakeholder간에도 요구 조정을 진행할 수 있다. Java 애플릿은 브라우저의 HTTP와는 독자적인 TCP/IP 통신을 수행할 수 있어서 브라우저로 이동된 Java 애플릿은 서버와 통신 채널을 유지한다. 또한, 요구 조정에 필요한 정보의 교환과 지속적인 세션 유지도 가능하고, 서버를 중개하여 요구 조정에 참가하고 있는 stakeholder와 데이터를 교환할 수 있다. 전송된 애플릿은 TCP/IP 프로토콜을 이용하여 서버와 지속적인 세션을 유지할 수 있으며, 여러 정보를 주고받을 수 있다. 이러한 Java의 특징을 이용하면 기존 WWW와의 상호운용성을 유지하면서, 이기종 플랫폼간에 특정한 전용 시스템없이 브라우저만으로 요구 조정을 진행할 수 있게 된다.

4.2 시스템 구조

동일한 시점에서 모든 stakeholder들의 데이터가 같은 내용을 유지하는 것은 필수적인 일로 동일한 데이터의 유지를 위해서는 데이터를 처리하는 응용 프로그램이 하나이거나, 혹은 여러 개의 응용 프로그램이 수시로 정보를 교환하는 방식을 취해야 한다. 응용 프로그램 공유방식은 (그림 5)와 같이 중앙 실행 방식과 복제 실행 방식으로 구분할 수 있다[22]. 본 요구 조정 시스템은 응용 프로그램이 각 브라우저로 전송되어 수행되나, 멀티포인트 세션 지원과 요구 조정 작업 세션

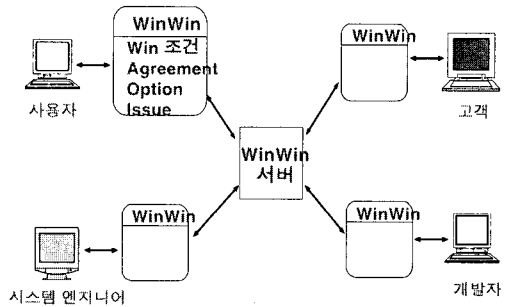
의 정보의 유지를 중앙의 서버가 담당하도록 기능이 분리되어 있어서 제안된 시스템은 기본적으로 복제 실행 구조의 모습을 가지고 있다. 그러나 요구 조정 작업의 데이터들은 중앙의 서버가 멀티캐스트하고, 사용자의 입력 전송과 데이터의 실제 처리는 참가자들의 브라우저내의 요구 조정 작업 응용이 처리해 주는 중앙 실행 방식의 형태도 일부 가지고 있어서 전체적인 구조는 복합 실행 방식을 취한다.



(그림 5) 요구 조정 작업의 실행 방식

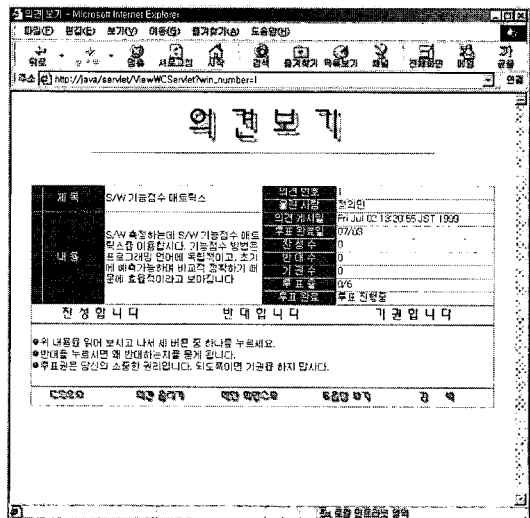
(그림 6)은 웹 기반 WinWin 요구 조정 구조의 개략적인 다이어그램을 보인다. 그림에서 보이는 대로, 각 stakeholder는 WinWin 클라이언트와 결합된다. 클라이언트는 WinWin 서버를 통해 통신한다. stakeholder는 자신들 개개의 win 조건을 정의하고, issues를 제기하고, option을 제안하고, 그리고 agreement를 유도하기 위한 WinWin 클라이언트 시스템 인터페이스와 상호작용 한다. stakeholder에 의해 생성되거나 갱신된 WinWin 객체는 WinWin 클라이언트에 의해 데이터베이스에 기록된다. stakeholder에 의해 수행되는 임의의 갱신 연산은 클라이언트에 의해 기록되고 다른 클라이언트에 통보되고 WinWin 서버를 이용하는 자신의 객

체 데이터베이스를 갱신하기 위해 이용된다. 그러므로 각 클라이언트는 필수적으로 모든 WinWin 객체의 사본을 유지한다. 제기된 issues는 stakeholder간의 조정 에 초점을 맞추어 이용된다. WinWin은 요구 조정을 지원하기 위한 도구를 제공한다. issue의 해결책은 win 조건에서 조정된 변경으로 귀결된다. 모든 issue가 해결되었을 때, agreement가 유도되고 시스템 요구 명세서를 정의한다.

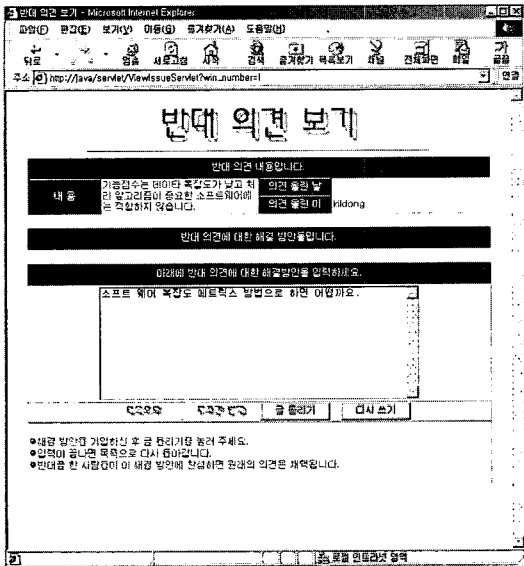


(그림 6) WinWin 요구 조정 구조

(그림 7)과 (그림 8)는 각각 웹기반 WinWin 요구 조정 시스템을 이용한 요구 조정 과정에서의 Win 조건 검색 과정과 Issue와 Option 검색 및 조정 과정을 보여준다.



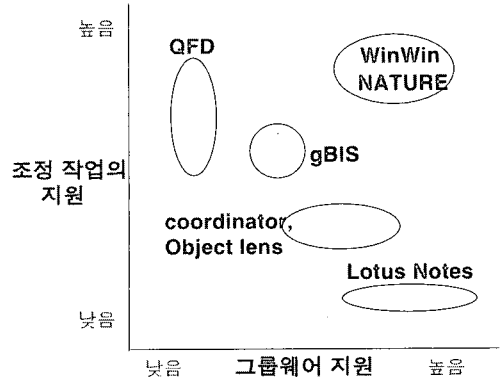
(그림 7) Win 조건의 검색



(그림 8) Issue와 Option 검색 및 조정

4.3. 운영 환경 및 성능 평가

(그림 9)는 그룹웨어 도구들을 요구 조정 작업의 지원 관점에서 비교한 결과이다[14]. 그룹웨어 도구 중에서, Object Lens[15]는 객체 지향 데이터베이스, 하이퍼 텍스트, 전자 메일, 그리고 조정 작업을 위한 “spreadsheet”을 제공하는 규칙 기반 지능형 에이전트로 구성되는 객체를 이용한다. Coordinator[16]는 전자 메일 시스템을 이용한다. Lotus Notes[17]는 공유 문서 관리기, 중복 알고리즘, 분산 협력 편집 시스템[18]을 지원한다. 이러한 도구는 요구 조정 과정에서 시스템 목표가 설정되고, 검토되고, 유추되는 구조를 제공하는데 실패하였다. 가장 널리 이용되는 Coordinator와 Notes의 경우에도 단지 제한된 메시지 전송 기능만을 제공한다. 이러한 도구는 주로 기술 중심이므로, 구성원간의 협력 작업을 위한 체계적인 기법이 결여되었다. 목표를 조정하는 기법의 gIBIS[19]는 시스템 목표간의 관계를 형성하기 위해 하이퍼텍스트 도구를 이용한다. QFD 기법[20]의 주요 문제 중의 하나는 다른 stakeholder가 소프트웨어 생명주기 후반에 작업에 참여할 때 잠재적인 충돌이 야기된다는 것이다. 또 다른 한계는 원래의 텍스트 기반 기법을 자동화하는데 대한 어려움이다. 조정 과정이 오프 라인으로 수행된다. NATURE 프로젝트의 Viewpoints[21]는 WinWin 요구 조정과 유사하지만 형식 모델을 제공하지 않는다.



(그림 9) 그룹웨어 도구들의 비교

일반적으로 공동작업 시스템의 성능평가를 정량적으로 계측하기는 곤란하다[22]. 마찬가지로 소프트웨어 요구 조정 시스템을 계량적으로 성능평가를 하기는 어렵다. 그 이유는 요구 조정 시스템의 성능은 실행 속도와 처리율로 평가하기 보다 사용의 용이성과 운영 환경에서 결정되기 때문이다. 따라서 본 논문에서는 다양한 형태의 인터넷 연결과 운영 체제를 가진 stakeholder들이 웹 브라우저만을 이용한 요구 조정 작업이 가능한지의 여부를 확인한다. 이를 위해 사용한 브라우저는 넷스케이프사의 Netscape와 MS사의 Explorer이고, 운영체제는 X-Window 환경과 Window 95 및 98 환경을 선택하였다.

LAN으로 직접 인터넷에 접속된 stakeholder간의 요구 조정 작업을 시험하기 위해 <표 1>과 같은 환경에서 테스트를 실시하였다. 모두 LAN으로 인터넷 접속이 가능한 환경을 테스트 장소로 선택하였으며, 웹 브라우저와 운영 체제는 혼용하여 테스트하였다. WinWin 요구 조정 서버는 교내에 위치하였다.

<표 1> LAN 연결시의 테스트 환경

	교내1	교외1	교내2	교내3	교외2
브라우저	Netscape	Explorer	Netscape	Explorer	Explorer
운영 체제	X-Window	Window98	Window95	Window95	Window95

테스트 결과 교내의 stakeholder들과 교외의 stakeholder들이 뚜렷한 속도의 차이없이 신속한 응답을 확인할 수 있었다. 그러나 시간대 별로 약간의 차이가 감지되나, 교내의 네트워크 상태에 기인한 것으로 판단된다.

다음으로 인터넷 접속 상황이 다른 stakeholder간의 요구 조정 작업을 테스트하였다. 테스트를 위해 이용된 실제의 프로젝트를 대상으로 소프트웨어 프로세스를 진행하였다. 이 테스트에는 총 5 팀의 stakeholder가 요구 조정 작업을 진행하였다. 2 팀의 stakeholder는 사설 인터넷 서비스 업체의 PPP 서비스를 이용하여 전화망으로 인터넷에 연결하였고, 나머지 3 팀은 LAN으로 연결된 인터넷상에서 요구 조정 작업을 시험하였다. 테스트 환경은 <표 2>와 같다.

<표 2> LAN 연결과 PPP 서비스를 이용한 테스트 환경

	LAN	PPP	LAN	LAN	PPP
브라우저	Netscape	Explorer	Netscape	Explorer	Explorer
운영 체제	X-Window	Window98	Window95	Window95	Window95

PPP 서비스를 이용한 stakeholder들과 LAN을 이용하여 인터넷에 연결한 stakeholder 모두가 유의할 만한 수준의 속도차이 없이 요구 조정 작업을 수행할 수 있었다.

앞의 두 가지 테스트를 통해 이질적인 환경의 인터넷 접속과 운영 체제에서 별도의 전용 시스템의 도입 없이 웹 브라우저만을 이용하여 성공적으로 소프트웨어 요구 조정 작업을 수행할 수 있었다. 이 결과로 이질적인 웹 브라우저, 인터넷 연결, 운영 체제, 플랫폼 아래의 다양한 stakeholder들이 WWW을 이용하여 전용 시스템의 지원이 없이도 소프트웨어 요구 조정 작업을 수행할 수 있는 가능성을 확인할 수 있었다.

5. 결론 및 연구과제

소프트웨어 프로젝트의 가장 중요한 성공 요소 중의 하나로 최근 인식된 것이 바로 소프트웨어 요구 조정이다. 소프트웨어 개발 프로세스의 모든 stakeholder의 효율적인 협력과 소프트웨어 요구조정은 성공적인 소프트웨어 프로세스를 위해서 중요한 전제 조건이다. 소프트웨어 요구 조정 시스템은 기본적으로 요구 공학(requirements engineering)에 적용될 수 있다. 목표는 모든 win 조건을 통합하고 win 조건의 충돌을 해결하여 일치시키기 위한 작업이다. 기존의 요구 조정 시스템은 전용 시스템 도입을 필요로 하며, 시스템이 설치된 장소에서만 요구 조정 작업이 가능하기 때문에 특정 플랫폼과 네트워크 환경에 종속되는 약점을 가지고

있었다. 또한 기존 시스템에서는 사용자가 참가자들의 네트워크 접속 상황을 사전에 인지해야 하며, 반드시 동일한 종류의 요구 조정 작업 응용 프로그램을 사용해야 했다. 따라서 본 연구에서는 이질적인 웹 브라우저, 인터넷 연결, 운영 체제, 플랫폼 아래의 다양한 stakeholder들이 WWW을 이용하여 전용 시스템의 지원이 없이도 소프트웨어 요구 조정 작업을 수행할 수 있는 웹 기반 소프트웨어 요구 조정 시스템을 개발하였다.

고전적인 요구 분석 기법은 사용자 요구만을 강조하고, 고객과 설계자와 같은 다양한 stakeholder로부터 요구를 획득하기에는 부족하다. 또한 stakeholder들의 다중 관점을 조정하기 위한 조정 이론이 제공되지 않는다. 그러나 웹기반 WinWin 요구 조정 시스템은 목표를 수립하고 관점을 통합하는 W 이론을 이용한다. 고전적인 요구는 주로 고정된 것에 비해 웹기반 WinWin 요구 조정 시스템의 win 조건은 조정 가능하다. 위험 분석은 초기의 수정을 제안하기 위해 명시적으로 처리된다. 이것이 나선형 모델에서의 요구를 점증적으로 개선하는 것을 용이하게 한다. 과거의 그룹웨어 도구는 소프트웨어 요구를 다룰 수 있는 구조가 결여되었기 때문에 요구 공학과 잘 연결되지 않았다. 그러나 웹기반 WinWin 요구 조정 구조는 하이퍼텍스트 형식으로 쉽게 처리될 수 있고 중요한 정보는 강조되어 웹 브라우저를 이용하여 검색될 수 있다.

WinWin 요구 조정 모델은 정형적으로 명세화되고 일관성을 위해 분석되었으나 이러한 과정에서의 가정에 관한 정확성과 유용성이 알려지지 않았다. 추후의 연구과제로는 우선 순위를 가지는 요구의 처리 및 인력 요인과 조정 과정간의 관계에 관한 모델링을 개선하고, 본 연구에서 제안한 웹기반 소프트웨어 요구 조정 시스템을 CSCW 시스템과 통합하여 완전한 공동작업 시스템을 구현하는 것이 남아있다.

참 고 문 헌

[1] B. Boehm, A. Egyed, J. Kwan, and R Madachy, "Developing Multimedia Applications with the WinWin Spiral Model," Technical Report TR-97-504, USC, 1997.
 [2] T. DeMarco, "The Role of Software Development Methodologies : Past, Current and Future,"

- Keynote Address, ICSE 18, IEEE/ACM, March, pp.2-4, 1996.
- [3] E. Yourdon, "Death March's Projects," Keynote Address, ICSE 97, IEEE/ACM, May 1997.
- [4] M. Weiser, "Software Engineering that Matters to People," Keynote Address, ICSE 97, IEEE/ACM, May 1997.
- [5] I. Sommerville, "Software Engineering," Fifth Edition, Addison-Wesley, 1996.
- [6] B. Boehm and R. Ross, "Theory W Software Project Management : Principles and Examples," IEEE Transactions on Software Engineering, 15 (7), pp.902-916, July, 1989.
- [7] C.A. Ellis, S. J. Gibbs, and G. L. Rein, "Groupware : Some Issues and Experience," ACM Communication, 34(1), pp.38-58, 1991.
- [8] B. Boehm, "A Spiral Model of Software Development and Enhancement," IEEE Computer, 21(5), pp.61-72, 1988.
- [9] B. Boehm and P. Bose, "A Collaborative Spiral Software Process Model Based on Theory W," Proceedings of International Conference on Software Process, pp.59-68, 1994.
- [10] B. Boehm et al., "Software Requirements As Negotiated Win Conditions," Proc. First ICRE, pp.74-83, 1994.
- [11] 김태훈, 신우창, 김태균, 신영길, 우치수, "분산 객체 지향 소프트웨어 개발 환경의 설계 및 구현", 정보과학회논문지(C), 3(2), pp.139-151, 1997.
- [12] B. Boehm et al., "Cost Models for Future Software Processes : COCOMO 2.0," Annals of Software Engineering, Vol.1, pp.57-94, 1995.
- [13] B. Boehm A. Egyed, J. Kwan, D. Port, A. Shah, and R. Madachy, "Using the WinWin Spiral Model : A Case Study," IEEE Computer, pp.33-44, July, 1998.
- [14] M. Lee and B. Boehm, "The WinWin Requirements Negotiation Systems : a Model-Driven Approach," Technical Report TR-96-502, USC, 1996.
- [15] K. Lai, K. Yu, and T. W. Malone, "Object Lens : A 'Spreadsheet' for Cooperative Work," ACM Transactions on Office Information Systems, 6(4), pp.332-353, 1988.
- [16] F. Flores, M. Graves, B. Hartfield, and T. Winograd, "Computer Systems and the Design of Organizational Interaction," ACM Transactions on Office Information Systems, 6(2), pp.153-172, 1988.
- [17] D. S. Marshak, "Lotus Notes, a Platform for Group Information Management Application," Patricia Seybold's Office Computing Report, 1991.
- [18] I. Grief and S. Sarin, "Data Sharing in Group Work," ACM Transactions on Office Information Systems, 5(2), pp.187-191, 1987.
- [19] J. Conklin and M. L. Begeman, "gBIS : A Hypertext Tool for Exploratory policy Discussion (graphical Issue Based Information Systems)," ACM Transactions on Office Information Systems, 6(4), pp.303-331, 1988.
- [20] S. Jacobs and S. Kethers, "Improving Communication Decision Making within Quality Function Deployment," 1st Intl. Conf. on Concurrent Engineering, Research and Application, August, 1994.
- [21] B. Nuseibeh, J. Kramer, and A. Finkelstein, "A Framework for Expressing the Relationships between Multiple Views in Requirements Specification," IEEE Transactions on Software Engineering, 20, pp.760-773, June, 1991.
- [22] 정의현, 박용진, "WWW상에서의 공동작업 시스템의 설계 및 구현", 정보과학회논문지(C), 3(4), pp. 384-396, 1997.

권기태



e-mail : ktkwon@knusun.kangnung.ac.kr

1986년 서울대학교 계산통계학과
이학사

1988년 서울대학교 계산통계학과
이학석사

1993년 서울대학교 계산통계학과
이학박사

1995년~1996년 미국 University of Southern California,
Post-Doc.

1990년~현재 강릉대학교 컴퓨터과학과 부교수

1999년~현재 한국전자통신연구원 초빙연구원

관심분야 : 소프트웨어공학, 데이터베이스, HCI