

시각적 언어를 사용한 소프트웨어 개발 프로세스에 관한 연구

- 건축 통합 솔루션 소프트웨어 개발을 중심으로

A Study of Software Development Method with Visual Presentation Language

- Case Study : Architecture Total Solution Software

김성곤(Sung-Kon, Kim)

울산대학교 디자인대학 디지털정보디자인학과

“이 논문은 2001년 울산대학교의 연구비에 의하여 연구되었음”

“This work was supported by University of Ulsan Research Fund of 2001 ”

1. 서론

- 1-1 연구 배경
- 1-2 연구 목적
- 1-3 연구의 범위와 방법

2. 시각적 언어 사용자 : 소프트웨어 디자이너

3. 시각적 언어 사용분야 : 사용자 중심의 소프트웨어

4. 개발 프로세스별 시각적 언어의 사용 사례

- 4-1 사례 연구의 개요
- 4-2 기반 기술 및 개발 영역 설정
- 4-3 컨셉 개발 및 개발 시스템 설정
- 4-4 사용자 및 사용환경 자료 수집
- 4-5 시나리오에 따른 요소 분석
- 4-6 주요 인터페이스의 개발
- 4-7 프로토타입의 개발

5. 사례 연구의 결과 분석

6. 결론

(要約)

최근 사용자 중심의 소프트웨어를 개발하기 위하여 소프트웨어 디자이너가 많은 역할을 하고 있다. 그들은 개발 요원 사이에서 서로 의사 소통의 도구로 사용되어질 시각적 문서를 제작하고 그리고 사용자 요구 사항을 분석하기 위한 시나리오에 기초를 둔 다이어그램을 제작한다. 본 논문에서는 건축 분야 실감형 서비스의 사례 연구 통하여 실질적인 예를 제시한다. 개발 영역의 설정, 주요 컨셉 개발 그리고 개발 시스템 설정의 개발 프로세스에서는 주요 진행과정과 개발원의 통일된 의견을 반영하기 위한 다양한 시각적 문서 제작의 예를 제시한다. 그리고 사용자 및 사용 환경 자료 수집과 시나리오 분석에 따른 요소 분석 과정에서는 소프트웨어 인터페이스를 제작하기 이전의 사용자 요구 사항 반영을 위한 디자이너의 시각적 언어를 사용한 문서화 제작의 예를 보여준다. 인터페이스 개발과 프로토타입 개발 과정에서는 개발원 사이에서 원활한 개발을 진행 위하여 다양한 시각적 문서의 필요 예를 제시한다. 이러한 시각적 언어를 사용한 문서들은 참여와 역할분배, 사용자 중심의 컨셉 개발, 책임 소재의 명료함, 다양한 인터페이스 개발 제안 등에 어떠한 중요한 역할을 하는가를 논한다.

(Abstract)

Recently, it is more important to do part of software designers for developing user-centered software. They produce visual document for helping communication between developers and visual diagram for analyzing user need factors based on user scenario. Case study of architecture real dispatch service can show developing samples. It is important to produce visual document for one combined opinion between developers in process of defining of development boundary, main concept development, selecting of development system. And visual document process with visual language by software designer is step for recognizing of user need factors before developing interface design in process of collecting design factors of user and environment and analyzing design factor by user scenario. Also, in process of interface and prototype development, it need visual document for peaceful and concord process between developers. Those visual documents with visual language could help to define developers role and duty, user-centered concept development and various interface development proposal during software developing.

<Keyword> Software designer, Visual language, User Centered

1. 서론

1-1 연구 배경

현재 소프트웨어의 사용은 특정한 부류의 사용자가 아닌 다양한 사용자층을 가지게 되었다. 이러한 이유로 소프트웨어의 개발에서는 기능 요소보다는 사용자 중심 요소를 더욱 중점을 두고 개발하게 되었다.

소프트웨어 개발에서 사용자 중심 요소를 조사하고 분석하는 능력은 소프트웨어 프로그래머 관련 개발자보다는 소프트웨어 디자인 관련 개발자가 보다 뛰어나다. 디자이너는 프로그래머보다 보다 거시적인 관점에서 소프트웨어를 조사하여 전체적인 사용자 요구사항을 만족시키려 노력하기 때문이다.



그림 1. 소프트웨어 개발 도구로서의 시각적 언어

이 논문에서는 소프트웨어 개발에서 디자이너가 사용자 중심의 요소를 어떠한 시각적 언어로 표현하여 프로그래머와 프로젝트 매니저에게 아이디어를 전달하고, 조사된 내용을 체계적인 문헌으로 정리하며 그리고 다양한 도표를 통하여 조사된 요소를 분석해나가는가에 대한 사례를 제시한다.

디자인 개발 사례의 프로세스는 한국전자통신연구소에서 개발되어진 분산형 실감 서비스를 위한 3D 멀티미디어 기술을 바탕으로 건축 통합 솔루션 소프트웨어의 개발 프로세스이다. 2년에 걸쳐서 개발되어진 기반 기술을 사용해서 사용자 중심의 어플리케이션 소프트웨어를 개발하는 연구이다.

1-2 연구 목적

본 연구를 수행함으로써 얻고자 했던 목적으로는 세 가지를 들 수가 있다.

첫째 : 시각적 언어의 주요 사용자인 소프트웨어 디자이너가 개발 과정에서 어떠한 역할을 가지고 관련자들과 함께 어떻게 진행하는가를 논한다. 그리고 다양한 소프트웨어에서 소프트웨어의 디자이너가 중심이 되어서 개발되어야 하는 사용자 중심의 소프트웨어의 종류와 그 특징을 논한다.

둘째 : 사용자 요구 분석은 소프트웨어 디자이너의 다양한 시각적 언어를 통하여 표현된다. 프로그래머, 매니저, 투자자, 그리고 그래픽 디자이너와의 의사 전달에 필요한 시각적 언어는 무엇이 있으며 그리고 사용환경 및 사용자의 요구의 체계적 문헌 기록, 통일성 있는 인터페이스 개발, 주요 기능 중심의 사용 프로세스 개발, 그리고 개발 기간의 단축에 용이한 시각적 언어들이 소프트웨어 개발에서 어떻게 사용 표현되어지고 있는가를 논한다.

셋째 : 본 사례 연구에서 나타난, 시각적 언어의 사용으로 인한 개발원의 정확한 의사소통과 소수의 개발원 위주가 아닌 전체 개발원의 적극적 참여 그리고 시각적 언어의 사용자 분석을 통한 소프트웨어 개발 성공 사례를 논한다.

1-3 연구의 범위와 방법

2장, 3장, 4장은 문헌 연구와 실제 개발사례 조사를 통해 이루어진다. 2장은 시각적 언어의 주요 사용자인 소프트웨어 디자이너의 각 개발 연구원과의 역할에 관하여 논한다. 3장은 소프트웨어 디자이너가 주요 개발 연구원으로 활동할 수 있는 소프트웨어의 종류에 관하여 논한다. 4장에서는 소프트웨어 디자이너의 실질적 시각적 언어를 사용한 표현 문서의 사례를 개발 프로세스별로 논한다. 5장에서는 사례연구를 통한 개발 결과의 분석으로 문서화의 장단점 및 향후 연구를 제시한다.

2. 시각적 언어 사용자 : 소프트웨어 디자이너

소프트웨어 개발이 과거 기능 중심에서 사용자 중심으로 변화함에 따라 사용자의 요구 사항을 이해하고 이를 개발에 반영할 수 있는 소프트웨어 디자이너가 등장하였다. 소프트웨어 디자이너는 과거의 소프트웨어 그래픽 디자이너와 인터페이스 디자이너와의 역할과는 다르다.

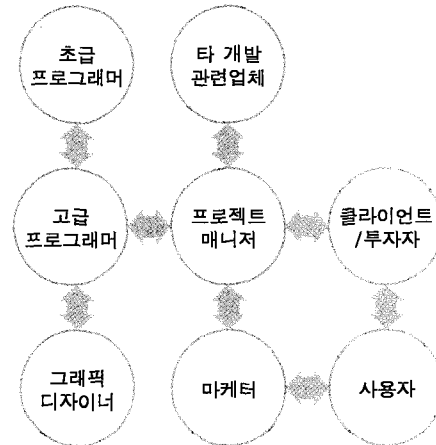


그림 2. 일반적인 소프트웨어 개발 관련자 관계

그림 2는 소프트웨어 디자이너가 없는 일반적인 개발 과정에서의 개발자들 사이에서의 관계이다. 고급 프로그래머와 프로젝트 매니저 그리고 투자자가 중심이 되어 개발되어지는데 일정한 문서에 의해서 서로의 스케줄 관리와 개발 범위 등을 설정한다. 그러나 조직원의 역할이 분화되는 소프트웨어 개발에서 개개인의 의견과 결과물이 전체적인 개발 과정과 틀린 경우가 있어서 개발비용이나 개발시간의 초과라는 문제점을 가지는 경우가 많다.

그림 3은 소프트웨어 디자이너가 개발 과정에서 중심이 되어 진행될 때 관련자들과의 관계이다. 소프트웨어 디자이너는 각 개발원들과 원활한 대화를 위하여 그리고 각 개발 시기마다 결정한 개발 방향에 대한 사항을 기록하기 위하여 시각적 언어를 사용한다. 일정표, 사용자 요구사항서, 샘플 데이터 분석서, 데이터 흐름 다이어그램, 소프트웨어 사용자의 가상 사용 다이어그램, 인터페이스, 그래픽, 그리고 사운드의 통일성을 위한 개발 외관 설정 문서, 그리고 개발 범위와 개발 자료의 수집 문서 등이 소프트웨어 디자이너 중심으로 시각적 언어를 사용해서 문서화된다.

소프트웨어 디자이너가 중심이 되어 개발이 진행되면 각각의 관련자들과 작업 내용의 문서화가 이루어진다. 소프트웨어 작업의 문서화는 개발 관련자들과의 전체적인 통일된 이해와 책

입 문제를 확실하게 할 수 있다. 다음은 소프트웨어 디자이너와 관련 개발자들과의 관계와 그 역할을 논한 내용이다.

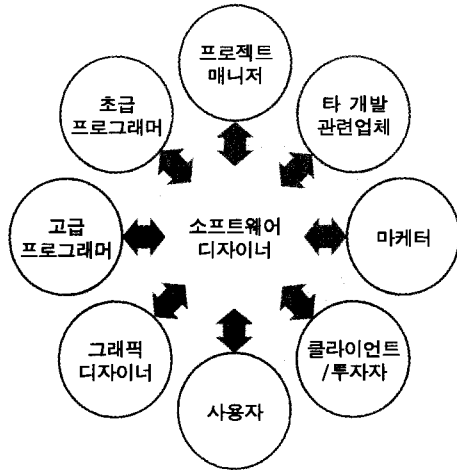


그림 3. 소프트웨어 디자이너 중심의 소프트웨어 개발 관련자 관계

• 프로젝트 매니저 - 소프트웨어 디자이너
개발의 일정관리를 전담하는 프로젝트 매니저는 프로그래머 혹은 경영자 출신이다. 이들은 디자인 관련 사용자의 요구 분석에 필요한 시간 정보에 약하다. 또한 그들의 경험을 통해서 작성되는 일정 관리에는 세부적 시스템 분석을 통한 일정 관리의 시각화가 부족하다. 소프트웨어 디자이너는 매니저의 일정을 모든 사람 각자의 역할과 개발 기간에 관한 정보를 정확히 전달 할 수 있는 문서화 작업에 도움을 줄 수 있다.

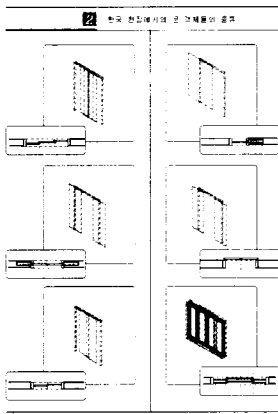


그림 4 한국형 창호의 종류 보고서

다. 그림 4는 AutoCAD사에서 한국형 ADT를 개발하고자 할 때 하청 받는 개발 업체에서 준비한 보고서이다. 한국형 창호에 대한 종류의 나열을 통하여 AutoCAD사는 하청업체가 개발할 내용을 시각적으로 이해할 수 있다.

• 마케터 - 소프트웨어 디자이너
소프트웨어 디자이너는 마케터에게 판매 촉진 및 제품 설명을 위한 자료를 제공 할 수 있다. 그림 5는 설계, 건축, 그리고 유지 보수의 일련의 과정 속에서 건축/자재 부분의 B2B 관련 소프트웨어를 개발하는 업체임을 설명하고 있다. 소프트웨어

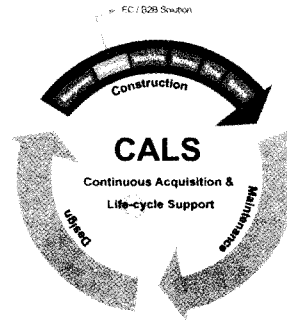


그림 5. B2B 솔루션의 개발 사업 영역 설명을 위한 다이어그램

디자이너가 개발한 이러한 다이어그램을 통하여 마케터는 구매자에게 해당 제품을 정확히 설명해줄 수 있다.

• 클라이언트 혹은 투자자 - 소프트웨어 디자이너
소프트웨어 디자이너는 투자자 확보를 위한 보고서에서 디자이너 고유의 스토리 중심 그리고 시각적 프리젠테이션 방식으로 참여한다. 이러한 보고서는 일반 투자자가 투자 대상에 대해 시간 내에 이해하기 쉽게 도와준다.

• 사용자 - 소프트웨어 디자이너
사용자의 요구 사항을 분석하는 과정은 소프트웨어 디자이너 역할 중 가장 중요한 역할이다. 또한 분석된 사용자 요구 사항을 시각적으로 나열하며 그리고 소프트웨어 개발 내용에 구체적으로 어떻게 적용되었는가를 시각적 다이어그램이나 표로서 설명한다.

• 그래픽 디자이너 - 소프트웨어 디자이너
그래픽 디자이너는 개발되어질 소프트웨어의 성격을 자세히 모른다. 소프트웨어 디자이너는 그래픽 디자이너에게 개발되어질 소프트웨어의 성격과 각 그래픽 디자이너의 그래픽 통일성을 유지하도록 도와준다. 이를 위하여 소프트웨어 디자이너는 그래픽 디자이너에게 대표되는 샘플을 시각적으로 제공하여야한다.

• 고급 프로그래머 - 소프트웨어 디자이너
고급 프로그래머는 소프트웨어 디자이너와 가장 의견 대립이 많은 사람이다. 프로그래머는 기능 위주의 소프트웨어 개발 및 그들의 개발 도중에 임의로 만든 인터페이스에 만족하는 경향이 있다. 소프트웨어 디자이너는 고급 프로그래머가 코딩 작업에 들어가기 전에 자세한 인터페이스 구조와 전체적인 데이터의 흐름 그리고 이러한 인터페이스가 왜 필요한지를 시각적으로 디자인하여 이해시킨다. 이 작업은 개발 기간의 단축과 가장 밀접한 관계를 가진다.

• 초급 프로그래머 - 소프트웨어 디자이너
초급 프로그래머들은 고급 프로그래머에게 필요한 코딩 지식을 교육받아서 비교적 단순한 프로그램 모듈을 코딩한다. 이들은 소프트웨어 디자이너가 시각적으로 표현한 전체적인 소프트웨어 구조, 인터페이스 그리고 데이터 흐름 등을 토대로 개발 소프트웨어에 대한 전반적인 지식을 짧은 시간 내에 습득한다. 이는 초급 프로그래머들이 범하기 쉬운 독단적이며 전체적 구조와 통일되지 않는 프로그램 모듈을 코딩하지 않게 도와준다.

소프트웨어 디자이너는 마케팅, 사용자분석, 프리젠테이션, 프로그램 코딩, 그래픽 그리고 인터페이스 설계에 대한 전반적인 지식을 갖추어야한다. 그러나 무엇보다도 각각의 개발 과정의 작업에서 모든 작업을 시각적인 언어로 문서화하는 능력이 가장 중요하다.

3. 시각적 언어 사용분야 : 사용자 중심의 소프트웨어

개발 과정에서 소프트웨어 디자이너가 시각적 언어로 많은 문서를 표현할 필요가 있는 소프트웨어는 기능 위주의 소프트웨어가 아니고 사용자 위주의 소프트웨어이다. 이런 소프트웨어의 종류들은 다음과 같다.

소프트웨어의 개발에 있어서 소프트웨어의 형태 분류의 기준으로 크게 두 가지를 들 수 있다. 첫째는 개발 시기에 필요한 지식 종류의 중요도이고, 둘째는 소프트웨어를 사용할 사용자의 특징에 따른 분류이다. 개발될 소프트웨어에서 개발자가 개발 도메인 지식(Domain-Oriented)이 많아야 하는가, 아니면 개발 시스템 지식(System-Oriented)이 많아야 하는가에 따라서 지식의 종류를 나눌 수 있다. 그리고 개발되어질 소프트웨어가 최종 사용자(End User)를 위한 소프트웨어인가, 아니면 소프트웨어를 통하여 다음 단계의 데이터를 만들어 내는 사용자(Develop User)를 위한 소프트웨어인가로 사용자를 나눈다.

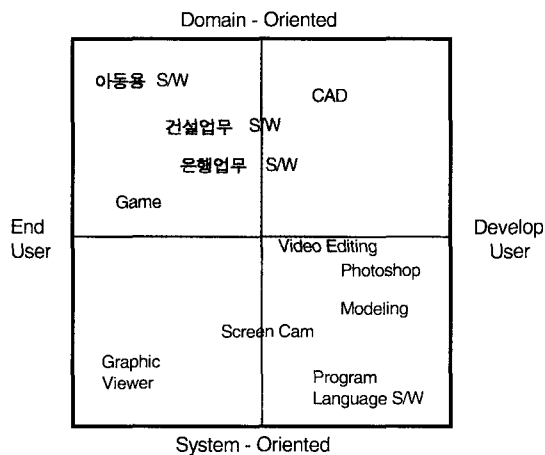


그림 6. 소프트웨어의 형태 분류

예를 들어, 게임 소프트웨어 개발에서는 해당 영역의 도메인 지식과 개발할 시스템에 대한 지식의 중요도가 비슷하게 요구되어진다. 또한 게임은 단지 최종 사용자에게 의해서 사용되어지고 다른 작업을 위한 과정으로 사용되어지지 않는다. 한편 모델링 소프트웨어는 개발영역의 지식보다는 필요한 기능을 구현하기 위한 컴퓨터 시스템의 전반적인 지식이 요구되어진다. 그리고 개발된 모델링의 결과는 다른 소프트웨어에서의 그래픽 자료로 사용되어질 가능성이 많다. 마지막 예로써 아동용 교육 소프트웨어는 전적으로 아동학의 전반적인 지식이 필요한 소프트웨어이다. 또한 아동들은 이 소프트웨어를 가지고 다른 작업을 위한 부산물을 만들어 낼 가능성은 적다. 이와 같은 여러 소프트웨어형태 중에서 사용자 중심의 개발이 필요한 부분은 그림 6.의 회색영역에 해당하는 도메인 지식이

중요한 요소로 적용되는 소프트웨어 형태와 최종 사용자를 위한 소프트웨어 형태이다. 이 소프트웨어들은 도메인 지식에 대한 체계적인 분석 및 중요 요소만의 추출 그리고 사용자의 일차적 요구 이면의 감성적, 인지적, 사회적, 문화적 요소 등의 분석들이 필요하다. 최종 사용자에게 의해 사용되는 소프트웨어들은 보다 인터페이스 개발과 시각적 그래픽 요소 및 청각적 음향효과에 개발에 중점을 두어야한다. 이 모두 시스템 기능적 소프트웨어 개발 접근보다는 사용자 중심의 소프트웨어 개발 접근이 보다 필요한 분야이며 또한 소프트웨어 디자이너가 개발 과정에서 시각적 언어를 사용한 많은 문서화 작업이 이루어져야한다.

다음에서는 사례 연구를 통해 사용자 중심의 소프트웨어 개발 각 단계에서 소프트웨어 디자이너가 어떠한 형태의 시각적 언어를 사용하여 그 역할(조사, 분석, 표현)을 하는가를 구체적으로 제시한다.

4. 개발 프로세스별 시각적 언어의 사용 사례

4-1 사례 연구의 개요

개발 사례는 한국전자통신연구원(ETRI) 컴퓨터 소프트웨어 연구소 멀티미디어부 입체 정보팀에서 개발되어진 분산형 실감 서비스를 위한 3D 멀티미디어 기술을 바탕으로 건축 통합 솔루션 소프트웨어 개발을 위한 연구이다. 14명의 연구진이 2년 동안 기초 기반 기술 획득, 주요 엔진 프로그래밍, 기반 기술을 적용할 소프트웨어 컨셉 개발, 그리고 구체적인 소프트웨어 개발 작업을 하였다. 이 개발을 위하여 프로젝트 매니저 1명, 보조 프로젝트 매니저 1명, 시스템 디자이너 2명, 고급 어플리케이션 프로그래머 2명, 초급 어플리케이션 프로그래머 3명, 고급 그래픽 엔진 프로그래머 2명, 데이터베이스 디자이너 1명, 그래픽 디자이너 1명, 그리고 소프트웨어 디자이너 1명이 참여하였다. 그리고 다수의 개발 분야의 도메인 전문가-자료 조사, 소프트웨어 테스트 그리고 자문역을 위한-가 개발 연구 팀에 파견되었다.

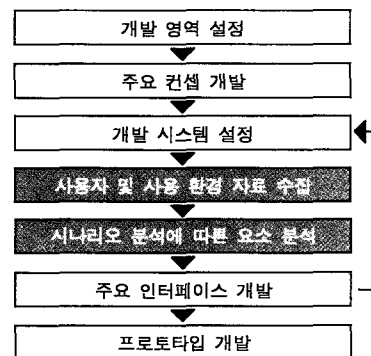


그림 7. 사용자 중심의 소프트웨어 개발 프로세스

본 사례 연구의 주요 프로세스는 그림 7.과 같다.1) 기반 기술

1) Joann T. Hackos, Janice C. Redish, "User and Task Analysis for Interface Design". 1998. p12

을 개발한 후에 적용할 소프트웨어를 개발하는 경우이기 때문에 먼저 개발 영역을 설정하였다. 건축의 적산, 내역 그리고 공정을 전반적인 일련의 과정을 포함한 건축 통합 솔루션 소프트웨어를 개발 영역으로 설정하였다. 그리고 다양한 컨셉을 개발하고 개발하고자하는 시스템을 설정한다. 그 후 사용자 및 사용환경자료 수집과 시나리오 분석에 따른 요소 분석의 과정을 걸친 후 주요 인터페이스를 설계한다. 그리고 세부적 모듈의 코딩작업에 해당하는 일차적 프로토타입을 제작하였다. 이 프로세스에서 시각적 언어를 사용하여 소프트웨어 디자이너가 더욱 주요 역할을 해야 부분은 사용자 환경 자료 수집과 시나리오에 따른 요소분석 과정이다.

4-2 기반 기술 및 개발 영역 설정

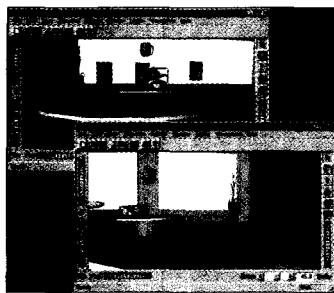


그림 8. 개발되어진 분산형 실감 기반 기술

개발되어진 기반 기술은 다수의 사용자가 가상 공간에서 서로의 의사를 전달하는 실감 서비스이다. 이 실감 서비스에서 사용자는 서로의 객체를 공유하여 조정할 수 있고 또한 3차원공간에서 기록할 수 있다. 부과적으로 화상 대화 및 채팅이 제공된다. 기반 기술의 어플리케이션 소프트웨어로써 건축 통합 솔루션이 설정되었다. 건축, 설비, 전기 등의 일련의 건설작업에 통합하여 다중 사용자 연결과 가상 현실 3차원 모델의 활용으로 사용자에게 보다 정확한 정보 전달과 각 일련의 작업을 단순화함에 목적을 두었다. 소프트웨어 디자이너는 개발 영역 설정에서 다양한 컨셉을 제

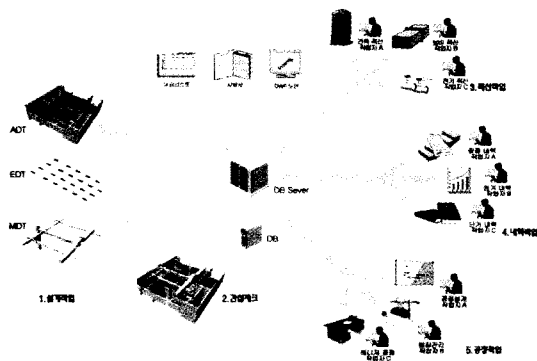


그림 9. 개발되어질 소프트웨어 영역 설명 다이어그램

시할 수 있으나 대부분의 경우 톱 매니저에 의해서 이미 영역 설정이 정해져 있다. 이때 소프트웨어 디자이너는 보다 구체적인 영역에 해당하는 시각적 문서화 작업을 하여야한다. 그림 9는 개발되어질 소프트웨어 영역을 구체적인 다이어그램으로 표현하였다. 컨셉 개발 회의에 참석하여 많은 의견을 나누었으나 실질적인 의견의 결정이나 전반적인 의견의 방향이 불확실한 경우가 있다. 이때 소프트웨어 디자이너는 전체적인 의견을 수렴할 수 있는 시각적 문서화 작업을 하여야한다.

4-3 컨셉 개발 및 개발 시스템 설정

개발 영역 설정은 대략적인 소프트웨어의 기능과 사용자를 정의한 반면, 컨셉 개발은 구체적으로 어떠한 전략을 가지고 사용자에게 소프트웨어를 사용할 때 시간 절약, 정확한 데이터의 공급, 안정된 시스템의 운영 등에 대한 컨셉을 개발하는 것이다. 이때 사용자 조사와 도메인 전문가의 의견을 많이 수용한다. 사례 연구로 개발되어진 소프트웨어는 무엇보다도 실행될 모듈의 종류와 순서 그리고 데이터의 흐름과의 연결이 컨셉 개발에 주요 내용이었다. 3차원 도면에 생성된 데이터들이 건축 물량 자동 산출, 자재 원가 데이터와의 연동, 공정 시간 스케줄과의 연결 그리고 3차원 분산 그래픽 데이터로의 전환의 방법을 해결해야 되었다. 그림 10은 각 팀장과 도메인 전문가와의 수 차례 회의를 통해 만들어진 다이어그램들 중의 하나이다. 소프트웨어 디자이너는 도메인 전문가가 제시하는 샘플 데이터, 팀 매니저의 개발 방향 그리고 고급 프로그래머의 개발 가능성의 의견을 참고로 시각적인 문서화 과정인 다이어그램을 개발한다. 그림 10은 개발되어질 소프트웨어의 5가지의 모듈 종류와 모듈들의 순서 그리고 데이터가 생성되는 순서 그리고 구체적인 샘플 데이터의 형태를 보여 주고 있다. 이후에 회의에 참여하지 않은 데이터베이스 디자이너를 포함한 개발 요원들이 전체적인 소프트웨어의 구조와 개발 방향에 이해하기 쉽게 만들어 준다.

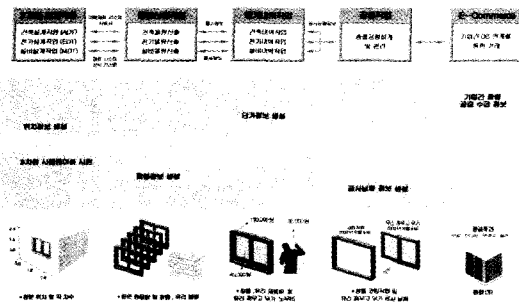


그림 10. 모듈별 데이터 생성 컨셉을 보여주는 다이어그램

개발 시스템 설정에서는 소프트웨어 디자이너는 그림 11. 같은 개발 시스템 구조도 다이어그램을 제작하여 개발원들의 이해를 돕는다.

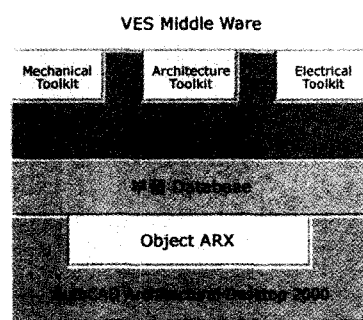


그림 11. 개발 시스템 구축 구조도

4.4 사용자 및 사용환경 자료 수집

사용자 환경 자료 수집은 데이터 자료를 체계적으로 요약하고 이를 시스템 디자이너, 프로그래머 그리고 데이터베이스 디자이너와 공유할 수 있도록 소프트웨어 디자이너가 주가 되어 작업을 진행하여야 한다. 방대한 데이터를 체계화하여 이를 시각적인 다이어그램이나 도표를 만드는 작업이다.

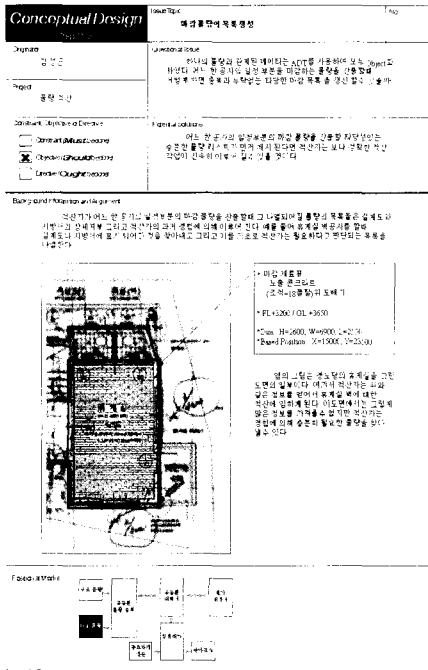


그림 12. 수집된 샘플 자료의 문서화 (2차원 휴게실 공사 도면)

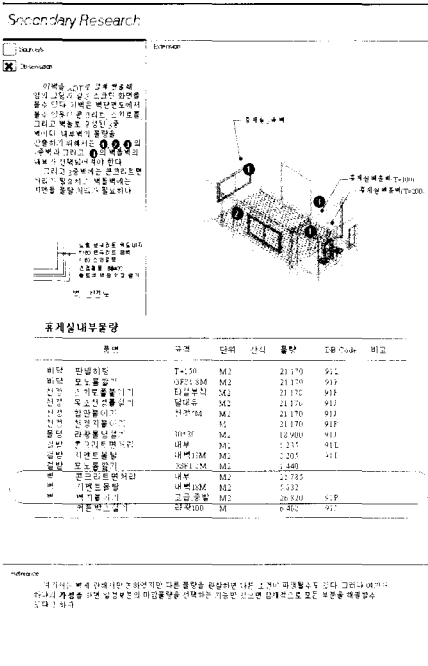


그림 13. 샘플자료에서 데이터 전산화를 위한 시각적 문서화

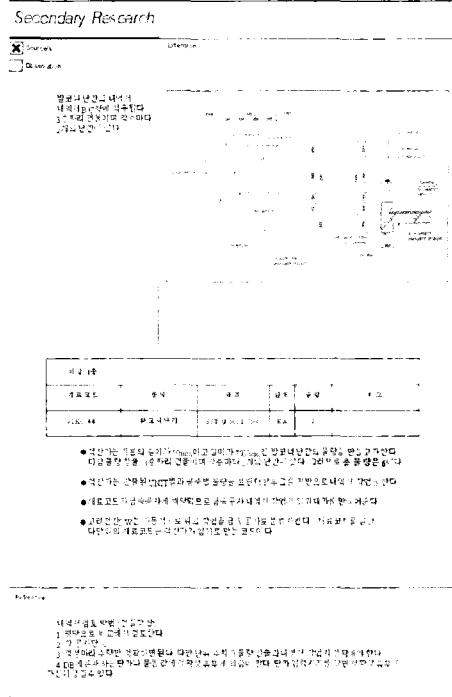


그림 14. 수집된 자료의 샘플 문서화 (금속 난간 공사 도면)

자료 수집에는 데이터베이스에 저장되어 응용 소프트웨어로 사용 되어져야 하는 디지털화된 데이터 수집과 개발을 위해 사용자와 환경의 전반적인 이해를 돕기 위한 데이터 수집으로 나누어진다.

그림 12은 2차원 도면에서 자료 수집의 샘플로 선택되어진 휴게실에 표시된 도면의 내용과 그 특징을 서술한 문서이다. 이 부분적인 도면에는 재료와 공사방법에 관하여 간략하게 기록 되어져 있다. 일반적인 건축 물량 산출자는 도면을 보고 건축 물량을 값을 산출한다. 개발 소프트웨어는 도면에 적힌 내용을 전산화하여 자동 건축 물량 산출 소프트웨어를 개발하고자 한다. 그림 13은 샘플 휴게실을 3차원 도면으로 어떻게 그려지며 그리고 3차원 도면에서 각 치수가 그림 12에 표시된 건축 재료의 치수로 적용되는가를 보여주는 예이다. 그림 12의 실제적인 데이터와 그림 13의 개발 되어져야 할 소프트웨어서 활용되어질 데이터의 형식은 다르다. 이를 어느 과정에서 어떻게 변화시킬 것에 관하여는 다음 과정에서 개발되어야 하고, 자료 수집 단계에서는 그 전산화된 데이터의 예를 보여 주어야 한다. 위의 그림과 같은 예를 작업실, 화장실 혹은 부엌 등의 여러 종류의 장소의 바닥, 창호, 벽, 그리고 도장 등에 관하여 예를 만들면, 다른 개발원들과의 대화의 기본 자료로서 혹은 보다 구체적인 인터페이스 요소들의 자료로서 활용되어진다. 소프트웨어 디자이너는 앞에서 행하여진 개발 컨셉과 개발 시스템의 충분한 이해가 있으므로 기본 자료를 도

2) Erich Gamma, "Design Patterns : Elements of Reusable Object-Oriented Software", October 1995. p221. 전체적인 데이터 구조를 처음부터 한꺼번에 설계되어질 수 없다. 먼저 주요 데이터 구조를 설명할 수 있는 간략한 샘플을 찾고 이를 중심으로 데이터 구조의 기초 작업을 행하고 이를 점차적으로 확대하여 나간다.

Secondary Research

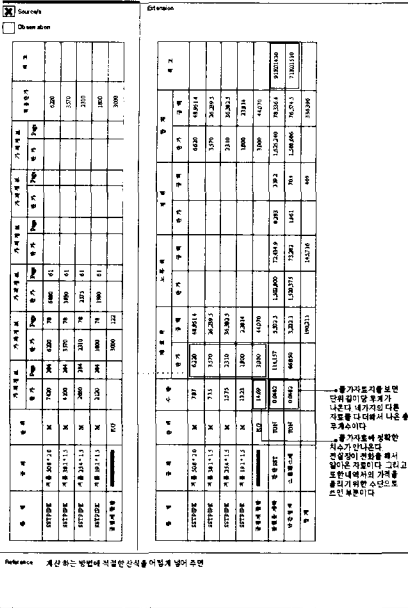


그림 15. 샘플자료에서 데이터 전산화를 위한 시각적 문서 II

메인 전문가와 대화하여 디지털화된 데이터로 유출할 수 있는 능력이 있다.

그림 14와 그림 15는 데이터 전산화의 다른 예를 보여 주고 있다. 앞의 그림 12, 13은 추상적인 도면에서 그 데이터 요소 추출과 적용과정을 보여 주고 있고 그림 14, 15는 구체적인 예의 공사에서 보다 자세한 데이터 생성 과정을 문서화시키고 있다. 그림 14는 아파트 난간의 금속 공사이다. 여기에는 각각의 치수가 자세히 존재하는데 이 치수들이 데이터 베이스의 테이블과 필드에 어떻게 적용 되는가를 보여 주고 있다. 이의 예는 데이터 베이스 설계자가 차후에 디자인 할 때 기본 자료로써 활용된다. 여기에는 각 난간의 치수와 각 난간 금속 파이프의 내역 원가가 적용되는 예를 보여주는데 도면, 물량 산출, 그리고 내역의 연결 방법의 실마리를 제시하여 주고 있다. 이와 같은 소프트 디자이너는 개발 도메인에서 자료를 수집하여 데이터를 전산화 할 수 있는 예를 시각적으로 문서화 시켜야 한다.

그림 16은 구체적인 전산화 데이터 생성이 아닌 사용자와 사용환경의 이해를 돕기 위해서 작성되는 시각적 문서이다. 여기에는 사용자의 특징과 관습적으로 행하여지는 원칙 그리고 사용 환경 요소로써 개발 소프트웨어를 사용할 사회 집단의 특징과 문화 등에 관하여 서술적으로 표현한다. 위의 시각적 문서는 사용자 중심의 소프트웨어를 개발하기 위하여 고려할 요소들을 나열하여, 향후 인터페이스 개발에 도움을 준다.

사용자와 사용환경의 데이터의 전산화된 시각적 문서와 사용자 중심 요소들이 서술된 시각적 문서를 통하여 다음의 과정인 시나리오 분석이 이루어진다.

3) Jacobson, Lvar; "Scenario-Based Design :Envisioning Work and Technology in System Development", 1995, p 85. Creating Contexts for Design

Scenario Analysis

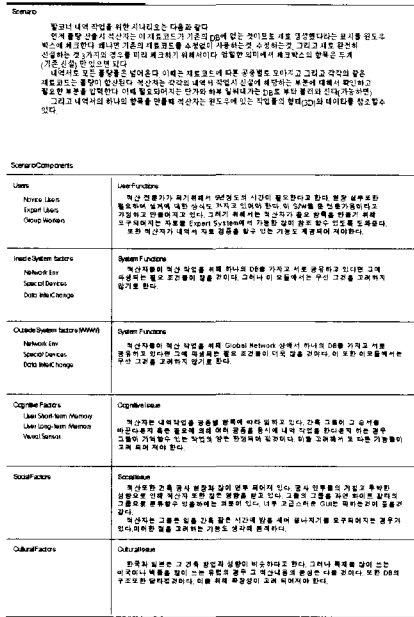


그림 16. 조사되어진 서술중심의 사용자 요소 문서화

4-5 시나리오에 따른 요소 분석

시나리오 분석에 따른 요소 분석 과정은 소프트웨어의 전체적인 모듈의 필요 기능 정의와 모듈들 사이의 관계 그리고 모듈들의 작업 순서 정의 등의 분석을 위하여 행하여진다. 이를 위하여 먼저 가상 시나리오가 대략적으로 작성되고, 이를 기반으로 그림 17. 같은 다이어그램이 만들어진다. 소프트웨어 사용 환경과 사용자 특징에 따라서 각 항목은 달라진다.

이 프로젝트에서는 사용자 입력모드(사용자의 키보드, 마우스, 혹은 다른 외부 기기를 통한 입력), 스크린 디스플레이(질문 윈도우의 팝업, 3차원 디스플레이 화면의 팝업, 데이터 자료 DBGrid 디스플레이, Location, Action, or Navigation, 계층구조 디스플레이), 보고서문(문자, 다이어그램, 그래픽파일), 사용자 멘탈 작업(Long-Term 기억, Short-Term 기억, 사용자 순간적인 판단, 사용자 지식에 바탕을 둔 긴 시간 사고 판단), 사용자 육체적 행위(건축 도면 찾기, 물가서 보기), 시스템 A.I.작업, 그리고 시스템 접근(사용자 로그, DB 접근, 네트워크 접근) 등이 시나리오 분석을 위한 항목으로 사용되었다. 그림 17은 프로젝트관리 및 샘플 공정 불러오기라는 소프트웨어 모듈에 관한 시나리오에 대하여 작성된 다이어그램으로써, 사용자의 행위와 소프트웨어의 기능과의 순차적 순서를 작성하며 그리고 데이터의 저장 시점에 관하여 표시하였다. 수십 개 혹은 수백 개의 시나리오 분석 다이어그램을 통하여 구체적인 소프트웨어 사용에 관하여 시각적 문서화를 만들었다. 이러한 문서에는 크게 데이터의 입출력 시점, 사용자의 멘탈 적용 시점, 그리고 시스템 접근 시점들이 표시되는데 이는 다음 과정의 주요 인터페이스 설계를 위한 기본 자료로 사용되어진다.

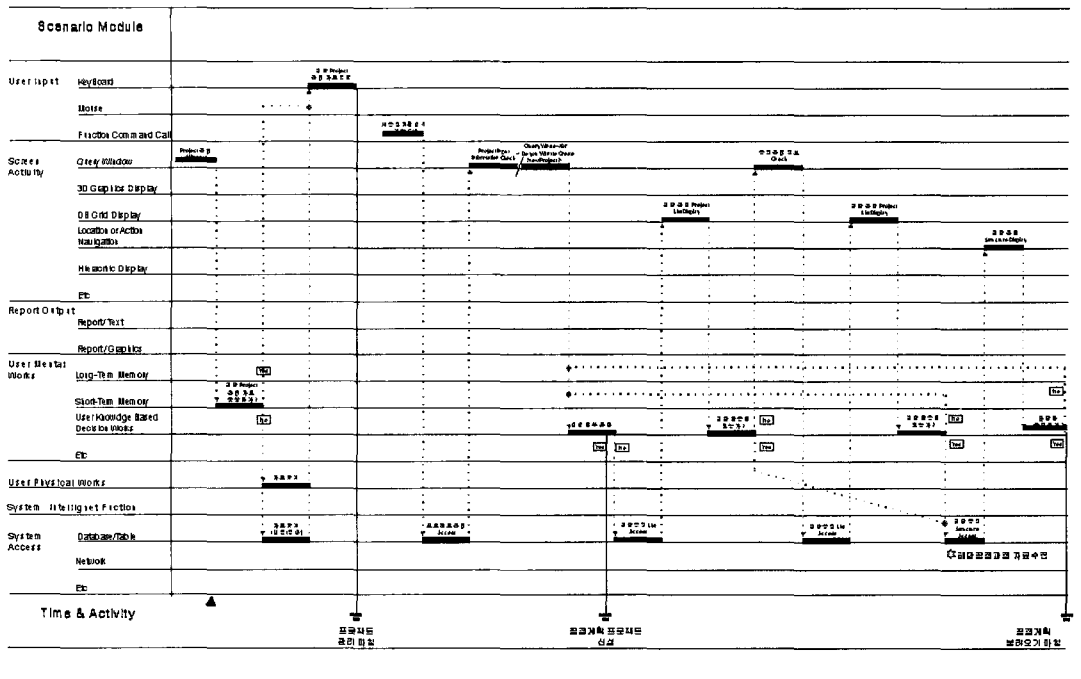


그림 17. 시나리오 따른 각 요소 분석 다이어그램

4-6 주요 인터페이스의 개발

주요 인터페이스 개발 과정은 위의 요소 분석을 바탕으로 소프트웨어가 가져야할 기능(Function) 함수들과 화면상에 디스플레이 되어질 아이콘 및 명령어들을 설정한다. 함수들과의 관계를 파악하여 삭제, 통합, 분할의 일련의 과정을 거치는데, 이와 같은 인터페이스 시각화 문서 작업이 끝나면 프로그래머들은 이를 바탕으로 프로그램 코드를 작성한다.

그림 17은 사용 환경, 사용자, 및 시스템에 관한 자세한 시나리오 항목들이 포함되어 있으나, 그림 18은 주요 인터페이스를 설계하기 전에 프로그래머가 이해하여야 할 기능 중심의 항목만 포함된다. 인터페이스 시각적 문서화 작업을 들어가기 전에 가장 큰 화면 레이아웃은 결정하여야한다. 그림 18을 보면 상단 좌측에 큰 레이아웃이 표시되어져 있다. 그리고 시나리오에 따라 주요 기능을 순서도 상에서 전개하여 나갈 때 주요 인터페이스 변화에 대한 장면 번호가 액티비티(Activity)의 박스 상단에 적힌다. 그림 18의 예를 들어 MRD001, RED001등과 같은 문자가 있다. 이는 각 모듈과 그 기능을 정의하여 붙여진 문자이다.

그리고 소프트웨어 디자이너는 그림 19에서 보는 봐와 같이 러프 아이디어 스케치와 각 기능들과 특징 그리고 데이터 등을 서술한 시각적 문서를 만든다. 그림 19는 MRD001의 예를 보여주고 있다. 다음 단계로 실제 사용되어진 프로그래밍

언어로 인터페이스를 구현한다.

그림 20은 MRD001 액티비티에 대한 그 예이다. 이 작업에서는 사용되어질 프로그래밍 언어의 특징을 소프트웨어 디자이너가 알아야 한다.

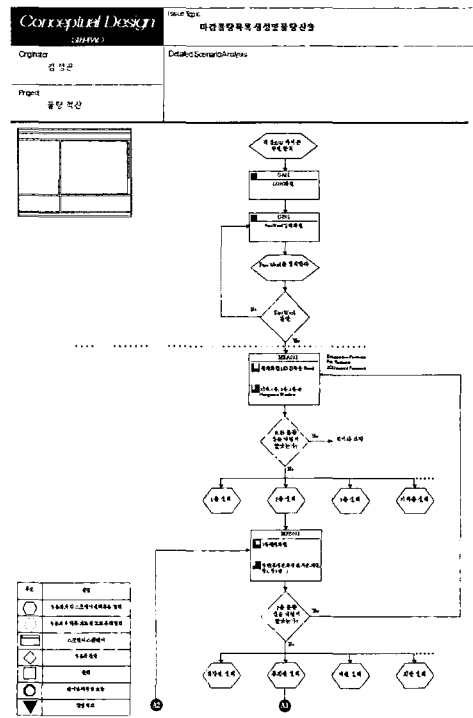


그림 18. 액티비티 순서도 다이어그램

4) Hugh Beyer, Karen Holtzblatt " Contextual Design : A Customer-Centered Approach to Systems Designs", 1998. p380. 개발 관련 데이터 정보를 처음으로 사용자 인터페이스 기준으로 표현하는 단계로서 소프트웨어 디자이너의 창조적이며 소프트웨어 개발 관련 전반적인 지식이 요구된다.

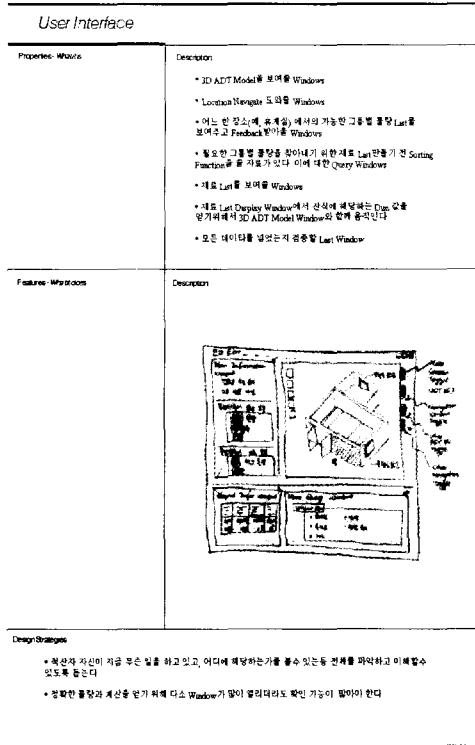


그림 19. 액티비터널 인터페이스 러프 아이디어 스케치와 설명

또한 프로그래머가 이러한 외관 인터페이스 제작한 파일을 복사하여 사용하기 때문에 프로그램 코드 속에서의 각 함수 명이나 변수 명의 정의가 사전에 약속이 되어있어야 한다. 그리고 프로그래머가 이해하기 쉽도록 실제 예를 보여줄 화면의 그래픽 작업을 하여야한다. 그림 20.에서는 건물의 콘크리트 구조는 아직 엔진과 접목이 안되어 표현이 불가능하다. 가상으로 이렇게 보여져야한다는 그래픽 파일을 제작하여 이해에 도움이 되도록 한다.

오브젝트(Object) 중심의 언어가 개발언어로 많이 사용되어진다. 그림 21.은 인터페이스 개발에서 새로운 오브젝트 개발이 필요하다고 고려되어질 때 소프트웨어 디자이너가 개발되어질 오브젝트의 특징을 시각적인 문서로 표현한 그림이다. 비주얼 베이직의 새로운 DBGrid OCX 개발에 관한 제안 문서이다.

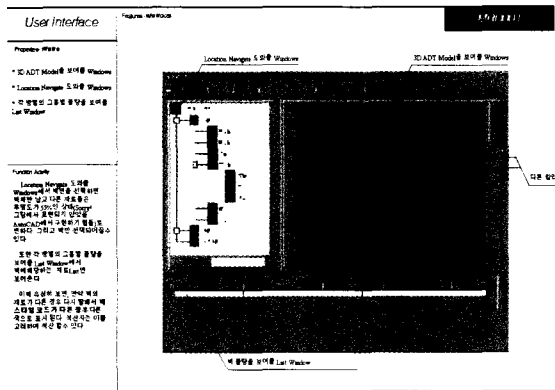


그림 20. 개발 프로그래밍 언어로 제작한 인터페이스와 기능 설명

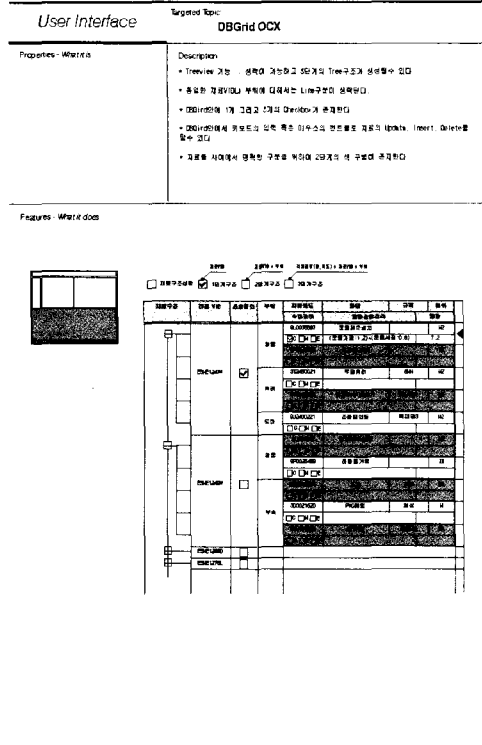


그림 21. 개발되어질 오브젝트의 인터페이스와 기능 설명

4.7 프로토타입의 개발

프로토타입 개발 과정에서 소프트웨어 디자이너는 두 개의 주요 역할을 가지고 있다. 첫째는 지금까지 작성한 시각적 문서의 요약 내용을 다이어그램으로 제작하여야한다. 둘째는 인터페이스 개발 과정의 확인이다.

매니저와 프로그래머는 많은 분석된 자료에 관심을 가지지 않는다. 그들은 역할에 필요한 요약된 자료가 필요하다. 그림 22.는 전체 여섯 개의 가장 상위 모듈중의 하나인 공정 모듈의 전개 요약 다이어그램이다. 그리고 그림23.은 실제로 제작되어진 프로토타입을 보여 주고 있다. 소프트웨어 디자이너는 자신이 개발한 의도대로 개발되어지고 있는가를 반복 확인 하여야한다. 때로는 프로그래머의 실수로 인하여 계획되어지지 않은 기능의 첨가나 인터페이스 변화가 이루어진다. 전체적인

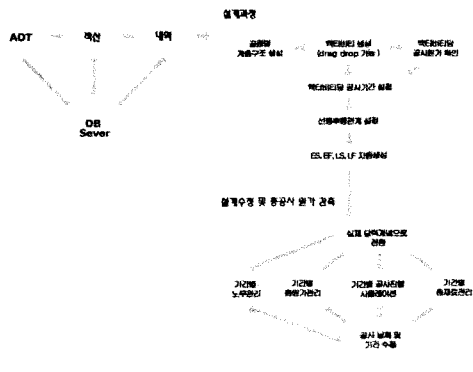


그림 22. 기능 요약 다이어그램

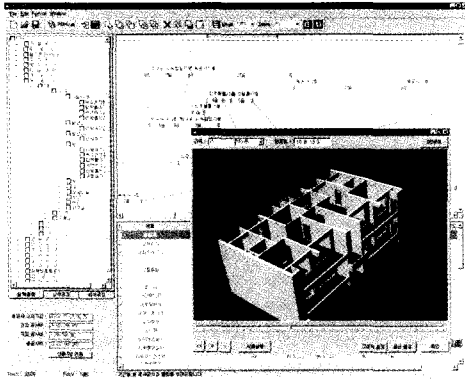


그림 23. 공정 모듈 프로토타입 개발의 예

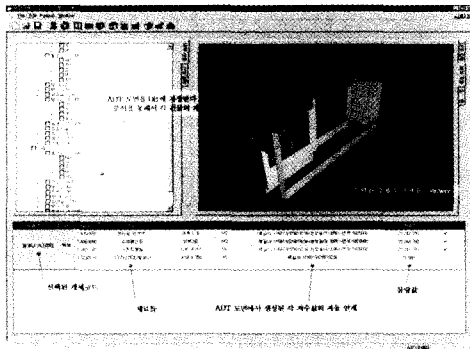


그림 24. 인터페이스 개발의 예2

소프트웨어가 가질 통일성과 간결성을 위반하지 않도록 반복 확인 하여야한다.

일차적인 프로토타입의 개발이 마친 후 소프트웨어 디자이너는 그림 24와 같은 화면을 캡처 받아서 각 기능을 설명한 시각적 문서를 만든다.5) 이는 프로토타입을 평가 할 전문가에게 보여질 문서로 활용되어진다. 직접 사용하여 보지 않고 짧은 시간 안에 전문가의 의견을 수렴하기 위하여 표현되어지는 데이터의 의미와 각 오브젝트의 특징을 시각적 문서로 표현한다.

5. 사례 연구의 결과 분석

사례 프로젝트를 수행함에 있어서 시각적 언어를 적용하여 사용자 요구분석 및 전반적 시스템 다이어그램 문서화를 진행한 결과 다음과 같은 결과 분석을 가진다.

• 개발 요원들의 역할 분배와 참여

연구 프로젝트를 수행한 후 제출되어진 개별적인 보고서에는 개발 요원들의 역할분배와 참여를 나타내는 요항들이 들어 있다. 특히 프로그래머들이 작성한 프로그램의 코딩 라인 수는 고급 프로그래머가 초급 프로그래머보다 적은 것은 일반적인 일이다. 그러나 불성실이 연구되어진 프로젝트에서 프로그램 코딩 라인 수는 고급 프로그래머와 일부 프로그래머에게 치중되어 있는 경향이 있다. 본 사례 프로젝트에서는 초급 프로그

5) Joann T. Hackos, Janice C. Redish, "User and Task Analysis for Interface Design", 1998. p391. 기능을 설명하는 시각적 문서 작업은 사용성 평가(usability testing)와 같은 반응(reaction)을 알기 위한 것이 아니고 세부적인 기능의 시각적 설명이 중심이 되어진다.

래머에게 전반적인 시스템 구조도와 사용자 요구 사항에 대한 이해도가 높았다고 보여진다. 시각적 언어를 사용한 문서화 작업은 모든 연구원들의 적극적인 참여 유도 할 수 있다.

• 사용자 중심의 컨셉 개발

정해진 개발 기간 2년 동안 시각적 문서화 과정이 비교적 많아서 프로젝트를 개발 기간은 단축되어지지 않고 동일하였다. 다만 개발 과정에서 시각적 언어를 사용한 사용자 요구 분석 과정이 많아서 충분한 사용자 중심의 컨셉 개발이 이루어 졌다. 특히 품질이나 고객 만족도가 높아서 재개발되어진 인터페이스 부분이 특히 적었다.

• 책임 소재의 명료함

개발과정에서 연구원들 사이에 있는 책임 월권행위와 무책임으로 인한 회의 시간의 연장 등의 경우는 현저히 줄어들었다. 소프트웨어 디자이너가 작성한 문서를 기초로 자신의 역할에 대한 분배가 확실하기 때문에 개발원들은 계획된 다른 팀원의 선행 과제의 불충분으로 인한 맡은 과제를 못 이행하는 경우가 드물었다. 특히 프로그래머와 인터페이스 개발자 사이의 의견 충돌은 현저히 줄어들었다. 왜냐면 매니저, 그래픽 디자이너 그리고 도메인 전문가 등의 의견을 조합하여 결정된 사항이므로 프로그래머는 이미 개발되어진 인터페이스의 실행 프로토타입 제작에 전력 할 수 있었다.

• 소프트웨어 디자이너의 직감적 인터페이스 개발

프로젝트 수행과정에서 이루어진 시각적 언어를 사용한 문서화 과정은 소프트웨어 디자이너와 인터페이스 디자이너에게 프로젝트를 이해하는데는 도움이 되었으나 너무 많은 양의 문서화 과정은 사용자에게 가장 적합한 인터페이스 제작하는 과정에서 문제점이 있었다. 인터페이스 디자이너에게 필요한 직감적 개발에 도움이 될 시각적 문서화 과정이 필요하다.

6. 결론

사용자 요소를 분석하는 능력을 가진 소프트웨어 디자이너가 소프트웨어 개발에 있어 시각적 언어를 이용한 문서화 작업에 관한 전반적인 과정을 분산 실감 서비스라는 사례 연구를 통하여 제시하였다. 이를 위하여 먼저 소프트웨어 디자이너의 역할과 각 프로젝트 개발원과의 관계를 논하였으며 그리고 소프트웨어 디자이너 개발 중심이 필요한 사용자 중심의 소프트웨어 종류에 관하여 논하였다.

각 개발 프로세스에서 개발 영역 설정, 주요 컨셉 개발 그리고 개발 시스템 설정에서 초기 개발원들의 원활한 대화 및 이해를 돕기 위한 다양한 다이어그램이 제시되었다. 이 과정에서의 문서는 향후 투자자 유치나 타 관련 제품 회사와의 홍보 및 제품 설명에 필요한 자료가 됨을 논하였다. 소프트웨어 디자이너에게 가장 필요한 사용자 및 사용 환경 자료 수집과정과 소프트웨어 개발에서 필요한 전산화 데이터 추출 문서 및 서술적 환경 문서에 관하여 논하였다. 인터페이스를 시나리오 중심으로 개발하기 위하여 시나리오 분석에 따른 요소 분석 다이어그램도 제시하였다. 기능(Function)과 동적(Dynamic) 데이터 흐름에 인지 공학적 요소를 함께 고려할 수 있는 '시나

리오에 따른 요소 분석 다이어그램'은 인터페이스 개발에 필요한 정보를 파악 할 수 있게 한다. 그리고 주요 인터페이스 개발 및 프로타입의 개발에서 프로그래머 코딩 이전의 단계의 문서에 관하여 논하였다. 마지막으로 사례 연구를 통하여 분석되어진 결과 분석을 크게 4가지에 관하여 논하였다.

본 연구는 사례 중심의 시각적 언어 문서화에 관하여 논하였다. 이러한 시각적 언어의 표현 문서는 사례별 그 차이가 크다. 전반적인 사례만을 제시하고 모든 사례에서 필요한 공통적 시각적 언어 표현 문서에 대하여서는 제시하여 주지 못하였다. 향후 이러한 시각적 언어 문서화 사례가 많아지면 대부분 소프트웨어 개발에 필요한 표준적 시각적 언어 표현 문서를 제시 할 수 있을 것이다.

시각적 언어의 표현 문서의 방대화는 인터페이스 개발 및 초급 연구원에게 혼란을 가져 준다. 문서 전체 체계의 간결한 구조와 인터페이스 및 초급 연구원에게 필요한 정보만을 전달 할 수 있는 문서 체계의 연구 또한 필요하다.

참고문헌

- . Bringing Design to Software , Terry Winograd; 1996
- . Contextual Design : A Customer-Centered Approach to Systems Designs", Hugh Beyer, Karen Holtzblatt
- . Design Patterns : Elements of Reusable Object-Oriented Software , Erich Gamma; October 1995
- . Designing the User Interface: Strategies for Effective Human-Computer Interaction , Shneiderman, Ben;1998
- . Designing Visual Interfaces , Kevin Mullet; 1995
- . Field Methods Casebook for Software Design , Wixon, Dennis ; 1996
- . Human-Computer Interaction , Alan Dix; 1998
- . Intelligent User Interfaces , Joseph W. Sullivan; 1991
- . Scenario-Based Design :Envisioning Work and Technology in System Development , Jacobson, Lvar; 1995
- . The Art of Human Computer Interface Design , Brenda Laurel; 1990
- . User and Task Analysis for Interface Design", Joann T. Hackos, Janice C. Redish
- . User interface design : Bringing the Gap from User Requirement to Design , Larry E. Wood, Boca Raton; 1998