제18회 한국정보처리학회 추계학술발표대회 논문집 제9권 제2호 (2002.11)

사용성 중심 설계에 의한 사용자 인터페이스 프로토타입 생성

김정옥*, 이창목*, 이정열*, 유철중*, 장옥배*
*전북대학교 컴퓨터과학과
e-mail:{jokim, cmlee, jylee}@cs.chonbuk.ac.kr
{cjyoo, okjang}@moak.chonbuk.ac.kr

Generating User Interface Prototypes by Usage-centered Design

Jeong-Ok Kim*, Chang-Mog Lee*, Jeong-Yoal*, Cheol-Jung Yoo*, Ok-Bae Chang*
*Dept. of Computer Science, Chonbuk National University

요 약

인터넷의 급성장으로 소프트웨어 환경이 웹으로 급속하게 전환함으로써 사용성 중심 설계가 새로운 패러다임으로 등장하고 있다. 본 논문에서는 사용자 중심 설계의 문제점을 보완하여, 사용성 중심 설계를 기반으로 효율적인 사용자 인터페이스 프로토타입의 생성과정을 제안하였다. 요구분석 시나리오를 기반으로 사용성 중심의 요구분석 프로세스를 개발하여 미숙한 설계자도 수준 높은 프로토타입을 개발할 수 있고, 프로토타입의 반복 횟수를 줄일 수 있게 하였다. 그리고 웹 환경에 적합한 사용성 중심의 사용자 인터페이스 프로토타입을 생성함으로써 통합 설계를 지원하기 위한 통합 모델링 언어(UML)를 지원할 수 있도록 하였다.

1. 서 론

인터넷의 급성장에 따른 소프트웨어의 환경이 웹으로 급속하게 전환함으로써 웹 애플리케이션이 새로운 패러다임으로 등장하게 되었다. 이러한 변화와 함께 사용자 인터페이스도 새로운 룰과 고품질의 입출력이 이루어지는 소프트웨어가 일반적이다. 또한 웹 애플리케이션의 성공은 사용자의 경험과 사용성에 의해서 결정된다. 그것은 고객이 찾고 있는 것을 찾을 수 없으면 고객은 떠난다는 것이다. 전문가인 Jackob Nielsen은 웹에서의 영업 잘못으로 수십억의 손실이 발생할 수 있는데 그 원인은 유용성 문제라고 추정될 수 있다고 평가하였다. 이러한 패러다임의 변화에 따라서 소프트웨어 엔지니어링의 개념도 사용자 중심 설계(user-centered design)에 대한 대안으로 사용성 중심 설계(usage-centered design)가 발전하게 되었다[1]. 사용자 중심 설계는 사용자를 이해하고 그들을 참여시키는 인간의 형이상학적인 측면의 기술요소를 통합하는 것이 미흡하다. 이러한 설계의 문제점을 보완하고 사용자 인터페이스를 효율적으로 하기 위하여 사용성 중심 설계가 이슈로 부각되고 있다. 따라서 본 논문에서는 요구분석 시나리오를 기반으로 요구분석 프로세스를 개발하여 미숙한 설계자도 수준 높은 프로토타입을 개발하고 프로토타이핑의 반복 횟수를 줄일 수 있도록 하였다. 또한 웹 환경에 적합한 사용성 중심 사용자 인터페이스의 설계를 지원할 수 있는 프로세스를 개발하여 프로토타입을 생성하는 방법을 제안하고자 한다. 그럼으로써 통합 모델링 언어(UML)를 지원할 수 있도록 하고자 한다. 본 논문의 구성은 2장에서는 사용자 중심 설계와 사용성 중심 설계의 차이점을 비교하고 사용성 중심 설계의 논리적 프로세스를 살펴본다. 3장에서는 사용성 중심 사용자 인터페이스의 프로세스의 설계 단계를 설명한다. 4장은 사용성 중심 사용자 인터페이스 프로토타입 개발 단계와 알고리즘을 제안하고, 5장에는 결론 및 향후과제를 기술한다.

2. 관련연구

2.1 사용자 중심 설계와 사용성 중심 설계의 비교

사용자 중심 설계와 사용성 중심 설계는 그림 1에서 보는바와 같이 상당한 차이를 볼 수가 있다. 사용자 중심 설계는 다음 세 가지의 기술을 구현하고 있다고 믿지만 실제로 도움은 줄지라도 이 기술을 효율적으로 지원하지는 못한다. 첫째는 사용자의 요망사항을 정확하게 정의하고 있다는 것이고, 둘째는 사용자 인터페이스 설계의 반복에 있어서의 피드백을 위한 빠른 문서 프로토타이핑(rapid paper prototyping)이 이루어진다는 것이며, 셋째는 프로토타입이나 시스템 작업에 있어서의 문제를 정의하기 위한 사용성 테스팅이 이루어진다는 것이다[2].

사용자중심설계 (User-centered design)	사용성중심설계 (Usage-centered design)
◆ 사용자에 초점 : 사용자의 경험과 만족	◆ 사용성에 초점 : 태스크성공을 지원하기 위한 개선된 도구
◆ 사용자 입력에 의한 운용	◆ 모델과 모델링에 의한 운용
◆ 실제적 사용자를 포함	◆ 선택적 사용자 포함
- 사용자의 연구	- 탐구적 모델링
- 설계의 참여	- 모델링의 확인
- 사용자 피드백	- 사용성 검사
- 사용자 태스팅	
◆ 반복적 프로토타이핑에 의한 설계	◆ 모델링에 의한 설계
◆ 매우 가변적, 비정형, 명기되지 않는 프로세스	◆ 체계적, 충분하게 명세된 프로세스
◆ 시운전과 에러에 의한 설계, 진화	◆ 공학적 설계

그림 1 사용자 중심 설계와 사용성 중심 설계의 비교

이것은 사용자가 함께하고 싶은 것과 진정으로 필요로 하는 것을 간과하기 쉽고 적절한 설계를 통해서 회피하려고하는 문제점을 발견하는 데에 있어서 비효율적으로 이루어진다. 웹에서의 사용성 중심 설계는 사용자와 성공적인 태스크를 위한 사용성에 초점을 맞추어 모델링이 이루어짐으로써 사용자가 떠나지 않도록 할 수 있는 통합적인 태스크 모델링이 요구되고 있다[3].

2.2 사용성 중심 설계의 논리적 프로세스

사용자 인터페이스 소프트웨어는 매우 크고 복잡할 뿐만 아니라 작성하기도 어렵고, 실행시 메모리도 인터페이스 측면에 집중되어있다. 특히, GUI 웹 환경에서의 인터페이스는 더욱 복잡하고 메모리의 효율성도 중요한 연구 분야라 할 수 있다. 웹 환경으로의 이전과 함께 사용성 중심 설계로의 변화로 더욱 사용자 인터페이스의 중요성이 부각되고 있는 상태에서 소프트웨어를 사용하기 쉽도록 보장해주는 기술이나 지침이 없기 때문에 소프트웨어에 대한 좋은 인터페이스를 구축하는데 어려움이 있다. 사용성 중심 설계를 위한 논리적 프로세스는 사용자인 인간 액터와 시스템 액터로 분류하고 역할, 태스크, 콘텐츠 모델은 서로 조화를 이루어 개발되고 다른 모델들과 연결되어지는데, 이 모델들은 도메인 모델, 비지니스 역할 모델, 작동 모델을 포함한다. 도메인 모델은 용어집과 보다 정교화한 데이터를 포함하고, 비즈니스 역할 모델은 애플리케이션의 기초가 되는 로직과 제약사항을 구체화하며, 작동 모델은 작동관계와 작업환경의 특징을 기술한다. 이벤트의 실제적 근원이 되는 액터는 사용자의 역할을 지원하는 태스크 케이스로 연결되고, 각 페이지와 폼 그리고 다른 상호작용 컨텐츠는 내부적으로 관련된 태스크 케이스의 묶음을 지원하는 추상화 프로토타입에 연결된다. 설계자는 태스크 케이스를 지원하기 위한 범위에서 특정 단계를 구현하는 추상화 컴포넌트로부터 행위 버튼, 링크, 테이블, 디스플레이와 다른 생성물을 추출한다[4]. 결국 이러한 태스크 케이스들은 사용자가 실행할 수 있는 역할을 지원한다.

3. 사용성 중심 프로세스의 개발 단계

3.1 예비단계

1) 본질적인 목적과 예상 : 비즈니스와 사용자 목적을 명료화 한다. 특징, 편리성, 내용물, 가능성을 예상하고 추측한다.
2) 예비 모델링 : 질문, 모호성, 위험하고 불확실한 분야를 정의한다.

3.2 프로세스 초기 개발단계

3) 역할 모델링 : 모든 사용자의 역할 목록을 작성하고 우선순위를 부여한다. 초기목표 서브 셋을 선정한다.
4) 태스크 모델링 : 모든 태스크의 목록을 작성하고 우선순위를 부여한다. 초기목표 서브 셋을 선정한다.
5) 태스크 클러스터링 : 유사성에 의한 모든 태스

크를 그룹핑 한다. 총체적인 네비게이션의 구조를 입안한다.

6) 기본 설계 : 상호작용과 가시적 골격을 입안한다. 심미적 설계를 검토하고 교정한다.
7) 추상화 프로토타이핑 : 서브 셋으로 선정된 태스크를 지원하는 상호작용 관계를 위한 콘텐츠를 개발한다.
8) 상호작용 설계 : 선정된 상호작용 관계를 위하여 상세화한 사용자 인터페이스 설계를 개발한다.
9) 가시화 프로그래밍 : 사용자 인터페이스로 설계된 부분을 프로그래밍한다

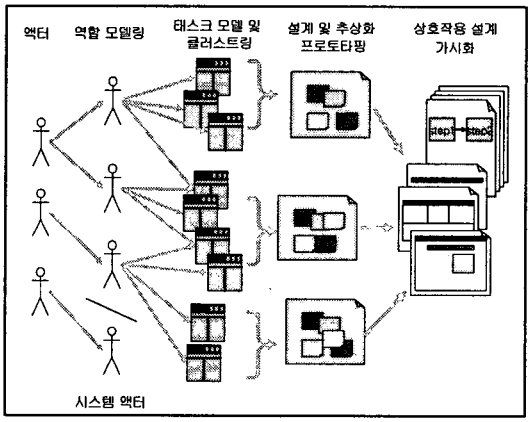

그림 2 사용성 중심 프로세스 개발 단계

3.3 성공적 프로세스 반복 단계

3) 역할 모델링 : 역할 모델을 검토하고 필요하면 교정한다. 다음 서브 셋을 선정한다.
4) 태스크 모델링 : 태스크 모델을 검토하고 필요하면 교정한다. 다음 서브 셋을 선정한다.
5) 태스크 클러스터링 : 네비게이션을 구조를 검토하고 필요하면 교정한다.
6) 기본 설계 : 상호작용과 가시적 골격을 검토하고 필요하면 교정한다.
7) 추상화 프로토타이핑 : 다음 선정된 태스크 서브 셋을 지원하는 상호작용 관계를 위한 콘텐츠를 위한 컨텐츠 모델을 개발한다.
8) 상호작용 설계 : 선정된 상호작용 관계를 위한 사용자 인터페이스 설계한다.
9) 가시화 프로그래밍 : 새롭게 설계 영역을 구성한다

4. 사용성 중심의 사용자 인터페이스 프로토타입 생성

4.1 사용자 인터페이스의 프로토타입 생성단계

1) 요구습득
역할 모델링 단계로서 Use Case Diagram (UseCaseD)을 작성하고, UsecCaseD의 use case를 위한 Collaboration Diagram(CollD)을 획득한다.
2) 시나리오로부터 특성분류
태스크 모델링 단계로 CollD-To-StateD 전환 알고리즘을 각 CollD에 반복적으로 적용한다.
3) 특성 분석
태스크 클러스터링 단계로 주어진 객체의 다양한 State Diagram(StateD)을 통합하고, 분석자는 동일 상태를 정의하고 공통 State명을 부여한다.

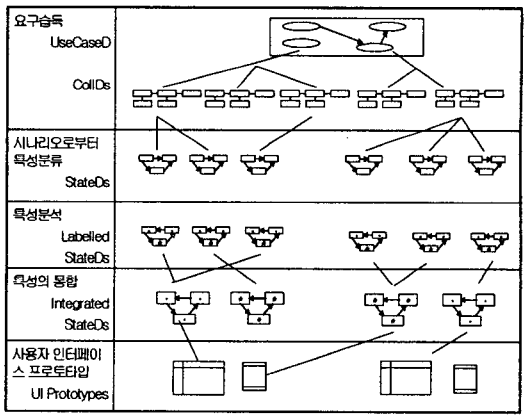

그림 3 사용자 인터페이스 프로토타입의 생성 단계

4) 특성의 통합
설계 단계로 객관적 활동에 의한 모든 부문의 StateD를 유스 케이스 마다 한 개의 StateD를 얻을 수 있도록 각 객체 유스 케이스를 통합한다.
5) 사용자 인터페이스 프로토타입 생성
추상화 프로토타이핑 및 설계 작성 단계로 시스템에서 발견된 모든 인터페이스 객체를 위한 사용자 인터페이스 프로토타입을 생성한다. 생성된 프로토타입은 다른 유스 케이스 사이에 연결하여 메뉴를 구성한다. 기초가 되는 인터페이스 객체의 상태로부터 동적인면과 정적인면의 사용자 인터

페이스 프로토타입이 생성된다. 프로토타입의 다른 화면은 객체의 정적인면을 가시화하고, 객체의 동적인면은 프로토타입의 대화 제어에 매핑된다.

4.2 사용자 인터페이스 생성을 위한 알고리즘

이 장에서는 4.1절에서 생성된 프로토타이핑 단계를 인터페이스 객체 행위로부터 프로토타입의 프로세스를 상세화 하고자 한다. 이 프로세스는 객체지향 언어로 알려진 "DOT" 표기법을 사용하여 다음 알고리즘을 요약할 수 있다.

```
Let IO be the set of interface objects in the system,
Let UC={uc1, uc2,..., ucn} be the set of use cases of the system,
For each io in IO
    For each uc in UC
        If  io.usedInUsecase(uci) then
            sd = io.getStateDforUsecase(uci)
            sd.generatePrototype()
        End If
    End for
    io.generateCompletePrototype()
End for
```

오퍼레이션 *usedInUsecase(uci)*는 객체 *io*에 적용하고, 만일 *io*가 하나 이상의 CollD를 포함하고 있는지를 체크하기 위하여 use case uc_i 연결한다. 만일 이 오퍼레이션이 *true* 이면 *getStateDforUsecase (uci)*이 호출되고 *sd*를 추출한다. 객체 *io* 행위를 캡쳐하는 State Diagram은 이 유스 케이스에 관련된다. StateD는 *sd* 로부터 사용자 인터페이스 프로토타입을 *generatePrototype()*을 사용하여 생성된다.

이 오퍼레이션 generateCompletePrototype()은 여러 유스 케이스를 위한 하나의 유일한 애플리케이션을 생성한 프로토타입으로 통합한다. 이 애플리케이션은 IO 객체가 있는 다른 유스 케이스를 옵션에 따라서 제공되는 메뉴로 구성한다.

5. 결론 및 향후 연구과제

사용자와 컴퓨터 사이의 상호작용에서 생성되는 데이터 플로우를 다루는 컴퓨터 프로그램이 사용자 인터페이스 부분이다. 또한 사용자 인터페이스 프로그램은 사용이 쉬워질수록 개발은 더욱 어려워진다. 이러한 애플리케이션 코드의 48%가 사용자 인터페이스에 그리고 구현 시간의 50%가 사용자 인터페이스의 구현에 소요된다는 연구가 나왔다[5]. 이러한 사용자 인터페이스 프로그램의 개발을 지원하기 위하여 본 논문에서는 사용성 중심 설계에 의한 사용자 인터페이스 프로토타입의 생성과정을 제안하였다. 이것은 요구분석 시나리오에 기반한 요구분석 프로세스를 개발하여 미숙한 설계자도 수준 높은 프로토타입을 개발할 수 있도록 하였다. 또한 웹 환경에 유용한 사용성 중심설계에 의한 사용자 인터페이스 프로토타입의 생성 프로세스를 개발함으로써 통합 설계를 위한 통합 모델링 언어(UML)를 지원할 수 있도록 하였다. 향후 연구과제로는 유스 케이스별 상호작용 객체들을 자동으로 그룹핑하고 배치할 수 있는 알고리즘을 개발하고, 시나리오로부터 유스 케이스 유형을 분류하여 사용자 인터페이스 유스 케이스 객체유형을 자동으로 생성할 수 있는 인터페이스 프로토타입의 생성에 관한 연구가 필요하다.

참고문헌

[1] L.L. Constantine, L.A.D. Lockwood, "Usage centered engineering for Web Applications" , IEEE Software vol. 19, issue 2, pp. 42-50, March-April 2002.

[2] M. Elkoutbi, I. Khriss, R.K Keller, "Generating user interface prototypes from scenarios", 1999. Proceedings. IEEE Inernati- onal Symposium on, pp. 150-158, 1999.

[3] C. stray, "TADEUS : seamless developement of task-based and user-oriented interfaces", stray C. Systems, man and cybernetics, part A, IEEE transactions on, vol. 30, issue 5, pp. 509-525, Sept. 2000.

[4] A. Knight, N. Dai, "Objects and the Web", IEEE Software vol. 19, issue 2, pp. 51-59, March-April 2002.

[5] M.D. Lozano, P. Gonzalez, I. Ramos, "User interface specification and modeling in an object oriented environment for automatic software development", Technology of Object-Oriented Languages and Systems, 2000. TOOLS 34. Proceedings. 34th international conference, pp. 373-381, 2000.