

## UI 프로토타입에 기반한 항해 설계

이옥진<sup>o</sup> 박상현 이병정\* 김희천\*\* 이숙희\*\*\* 우치수  
서울대학교 컴퓨터공학부, 서울시립대학교 컴퓨터과학부\*,  
한국방송통신대학교 컴퓨터과학과\*\*, 서경대학교 인터넷정보학과\*\*\*  
{duri96<sup>o</sup>, zez4shy, wuchisu}@selab.snu.ac.kr, bjlee@venus.uos.ac.kr\*,  
hckim@knou.ac.kr\*\*, suk@skuniv.ac.kr\*\*\*

### Navigation Design Based On UI Prototype

Sanghyun Park<sup>o</sup> Wook-jin Lee ByungJeong Lee\* Heechern Kim\*\* Sukhee Lee\*\*\* Chisu Wu  
School of Computer Science & Engineering, Seoul National University  
School of Computer Science, University of Seoul\*  
Dept. of Computer Science, Korea National Open University\*\*  
Dept. of Internet Information, Seokyeong University\*\*\*

#### 요약

웹 응용 관련 기술이 널리 이용되면서 적은 비용, 짧은 개발 기간, 고품질 등 새로운 요구사항이 등장하고 있다. 웹 응용 개발 현장에서는 이러한 요구사항을 만족시키기 위하여 개발 초기부터 UI 프로토타입(UI Prototype)을 적극적으로 이용하는 방법을 택하고 있다. UI 프로토타입에는 클라이언트 측의 항해가 구현되어 있기 때문에, 항해 설계 활동(Navigation Design Activity)이 불필요하다고 생각할 수 있다. 그렇지만 UI 프로토타입의 복잡성과 비정형성 때문에 큰 규모의 웹 응용 개발 과정에서 요구사항을 만족시키지 못할 가능성도 있다. 이 논문에서는 항해 모델을 이용하여 웹 응용 전체를 쉽게 파악할 수 있고, 사용자 관점 및 서버 관점에서 항해의 문제점을 파악할 수 있다는 점을 근거로 UI 프로토타입에 기반한 항해 설계의 필요성을 제시한다. 그리고 웹 응용 개발 프로세스에서 항해와 관련한 활동들의 세부 단계를 밝히고, UML SPEM 프로파일(Profile)을 이용하여 세부 단계를 표현한다.

#### 1. 서론

웹 기술은 이제 조직 내부의 업무 시스템에까지 널리 채택되고 있다. 자연스럽게 웹 응용 개발 과정에서도 기능과 관련한 요구사항 외에 적은 비용, 짧은 개발 기간, 고품질 등의 제약 사항을 고객들이 요구하고 있다. 웹 응용 개발 현장에서는 이러한 요구사항을 만족시키기 위하여 개발 초기부터 UI 프로토타입(UI Prototype)을 적극적으로 이용하는 방법을 택하고 있다. 이런 경우 항해 설계 활동(Navigation Design Activity)이 불필요하다는 시각도 있지만, UI 프로토타입의 복잡성과 비정형성 때문에 큰 규모의 웹 응용을 개발하는 경우 오히려 요구사항을 만족시키지 못할 가능성도 있다. 때문에 항해 설계 활동의 필요하다는 시각도 있다. [5]. 이 논문에서는 항해 설계 활동의 필요성에 대해 자세히 논의한다.

기존의 많은 연구는 클라이언트에게 보이는 결과만을 대상으로 항해 정보를 표현했다. [1, 2, 3, 4] 그렇지만 웹 응용의 업무 논리 계층(Business Logic Layer) 등 서버 측 항해 정보를 포함하면 웹 응용을 더 깊이 이해할 수 있다. 즉 항해 정보에 사용자 인터페이스(UI)와 업무를 처리하는 컴포넌트에 대한 정보를 포함함으로써, 웹 응용 설계자, 개발자, 그리고 고객까지 더 빠르고 쉽게 지금 개발하고 있는 웹 응용을 이해할 수 있다. [5] 이 논문에서는 이처럼 확장된 항해 개념을 채용한다.

웹 공학 환경(Web Engineering Environment, 이하 WEE로 표기)에서 언급하고 있는 WEE 프로세스는 설계 및 UI 정제 반복(Design and UI Refinement Iteration)을 정의하고 있다. 이 반복의 여러 활동 가운데 항해 설계 활동(Navigation Design Activity), UI 프로토타입 정제 활동(UI Prototype Refinement Activity)이 항해와 관련되어 있다. 이 활동들을 통하여 이전 반복에서 작성한 프로토타입으로부터 항해 정보를 추출한다. [5]

추출한 항해 정보는 검증하고, 수정한다. 이 같은 활동을 통하여 요구사항에 맞는 항해를 구현할 수 있다.

논문의 구성은 다음과 같다. 2장에서는 항해 설계의 필요성에 대해서 논의하고, 3장에서 세부 절차를 밝히고, UML Software Process Engineering Metamodel(SPEM) 프로파일 [6]을 이용하여 정리한다. 4장에서 결론 및 향후 연구 방향을 밝힌다.

#### 2. 항해 설계의 필요성

항해 설계는 프로토타입의 비정형화된 항해 정보를 체계적으로 표현하는 것이다. 프로토타입을 채택한 기존의 연구들은 설계를 마친 다음 프로토타입을 제작하는 방식을 택하고 있고, 프로토타입은 설계 산출물인 항해 모델보다 완성된 웹 응용에 가깝다고 볼 수 있다. [4, 7] 그래서 WEE 프로세스에서 개발 초기부터 고객과의 활발한 의사소통의 도구로 사용하기 위하여 사용한 프로토타입을 완성한 후, 다시 항해 설계를 한다는 것이 시간과 비용을 낭비하는 활동으로 보일 수 있다. 그럼에도 불구하고 항해 모델을 이용하여 몇 가지 이득을 얻을 수 있다.

첫째, 설계 산출물인 모델을 이용하여 프로토타입 보다 쉽게 웹 응용 전체를 파악할 수 있다. 프로토타입은 고객이나 UI 개발자에게는 친숙한 형태지만, 웹 응용의 전반에 걸친 항해 흐름이나 페이지 사이의 관계를 나타내는 것은 불가능하기 때문이다.

둘째, 모델을 이용하여 사용자 관점에서 항해의 문제점을 파악할 수 있다. 웹 응용의 복잡도가 증가할수록 프로토타입과 같이 정형화되지 않은 산출물을 이용해 끊어진 링크를 찾는 데가, 프레임 안에서 자기 자신을 호출하는 등의 항해의 문제점을 찾기 힘들어진다. 따라서 정형화된 모델을 이용하여 이

문제를 해결할 수 있다.

셋째, 서버에서 진행되는 항해의 문제점을 파악할 수 있다. 재사용 가능 컴포넌트를 사용하기로 결정된 항해의 경우에 컴포넌트의 인터페이스는 이미 알려져 있는 상태이다. 이를 토대로 사용자가 서버의 컴포넌트가 필요로 하는 파라미터를 모두 보내는지, 컴포넌트는 결과 페이지가 요구하는 파라미터를 모두 돌려주는지 확인하는 것은 웹 응용 개발에서 중요한 작업이다.

이러한 이유로 프로토타입이 완성된 후에도 항해 설계 활동을 통하여 항해 모델을 추출하는 작업이 필요하다.

### 3. 항해와 관련한 활동들의 세부 절차

항해 설계 활동과 UI 프로토타입 정제 활동은 설계 및 UI 정제 반복에 속한 활동이다. 항해 설계 활동은 UI 프로토타입, 소프트웨어 요구사항 명세(Software Requirements Specification) 등의 산출물을 이용하여 항해 모델을 만들어 내고, UI 프로토타입 정제 활동은 항해 모델 등의 산출물을 이용하여 UI 프로토타입을 정제한다.

그림 1은 설계 및 UI 정제 반복의 활동도 중 일부이다. 항해 설계와 관련하여 UI 설계자와 컴포넌트 개발자가 협력하는 것을 나타내고 있다. 구체적으로 컴포넌트 개발자가 수행하는 항해 설계 활동은 UI 설계자가 만들어낸 UI 프로토타입을 이용하고, 동시에 항해 설계 활동의 산출물인 항해 모델은 UI 프로토타입 정제 활동의 입력 자료로 쓰인다.

그림에 사용한 표기는 UML SPEM 프로파일에 따르면 있는데, 이것은 소프트웨어 프로세스를 UML에 기반하여 표기하기 위해 Object Management Group에서 정의한 프로파일이다.

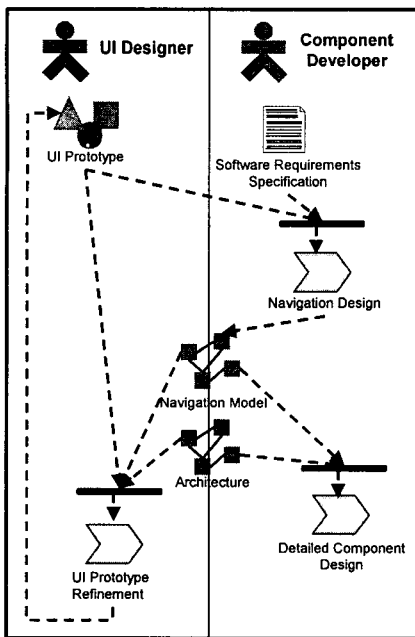


그림 2 항해 설계 활동(Navigation Design Activity)을 포함하는 설계 및 UI 정제 반복(Design and UI Refinement Iteration)의 활동도 일부

이 그림에서 사용한 표기는 부록에 자세히 정리했다.

### 3.1 항해 설계 활동 세부 절차

항해 설계 활동의 세부 절차는 다음과 같이 정리할 수 있다.

첫 단계에서 UI 프로토타입에서 모델을 추출한다. 모델을 추출하기 위하여 사용자 페이지에서 분석해야 할 부분은 메타 요소(Meta Element), 앵커(Anchor) 요소, 폼(Form) 요소, 인풋(Input) 요소 등과 자바스크립트(Javascript) 등의 스크립트이다. 그리고 서버 페이지에서 재전송(redirect) 등 항해와 관련된 부분도 추출해야 한다.

항해와 관련한 컴포넌트도 모델에 추가할 수 있다. 재사용 가능 컴포넌트의 경우, 프로토타입을 제작할 때, 주석을 이용하여 별도로 정보를 제공할 수 있다. 예를 들어 서버에 제출(submit)하는 폼 요소 내부에 이 폼 요소의 전송을 처리하는 컴포넌트의 식별자를 포함한 주석을 추가하여 모델 추출에 도움을 줄 수 있다. 그림 2는 이러한 주석의 예를 보여준다.

```
<form name="find" method="post" action="bbs_list.asp"
      onSubmit="return checkform(this)" ID="Form2">
      <!-- @component_id="SNU_SELab_CP2005024" -->
      ...
</form>
```

그림 1 서버에 전송되는 폼 요소(Form Element)를 처리할 컴포넌트 식별자를 주석에 표시하는 예

두 번째 단계로 추출한 모델을 소프트웨어 요구사항 명세를 이용하여 보완한다. 기본적으로 프로토타입은 재사용 가능 컴포넌트와 관련한 정보나, 항해와 관련한 선조건, 후조건을 모두 반영하고 있다고 가정한다. 그러나 중요도를 감안하여 한번 더 확인하는 절차를 둔다.

세 번째 단계는 모델 검증이다. 사용자 관점에서 바라 본 항해의 문제점 및 서버 관점에서 바라 본 항해의 문제점이 존재하는 지 확인하는 단계다. 사용자 관점에서 바라 본 항해의 문제점의 예로 끊어진 링크, 프레임 안에서 자기 자신을 호출 등을 들 수 있다. 그리고 서버 관점에서 바라 본 항해의 문제점의 예로 사용자가 서버의 컴포넌트가 필요로 하는 파라미터를 모두 보내는지, 컴포넌트는 결과 페이지가 요구하는 파라미터를 모두 돌려주는지 등을 들 수 있다.

구체적으로 컴포넌트에서 요구하는 파라미터 내역을 알고 있다면, 폼 요소가 가지는 인풋 요소의 이름과 비교하거나 세션(session)에 저장된 정보의 이름과 비교하여 컴포넌트가 요구하는 정보를 모두 제공하는 지 여부를 확인할 수 있다. 인풋 요소가 제공하는 정보는 타입을 알 수 없지만, 세션에서 얻은 정보를 컴포넌트에 제공할 경우, 타입도 비교할 수 있다. 물론 클라이언트 페이지에서 주석을 이용하여 인풋 요소의 타입 정보를 제공할 수 있다.

결과를 표시할 서버 페이지는 컴포넌트가 돌려주는 결과와 세션에 저장된 정보를 이용한다. 따라서 컴포넌트가 돌려주는 정보와 세션에 저장된 정보의 내역 및 타입을 이용하여 모델을 검증할 수 있다.

네 번째 단계에서 모델에서 발견한 오류를 수정하고, 항해 설계 활동을 마무리한다. 이상의 세부 단계를 UML SPEM 프로파일의 표기를 이용하여 다음 쪽의 그림 3과 같이 정리할 수 있다.

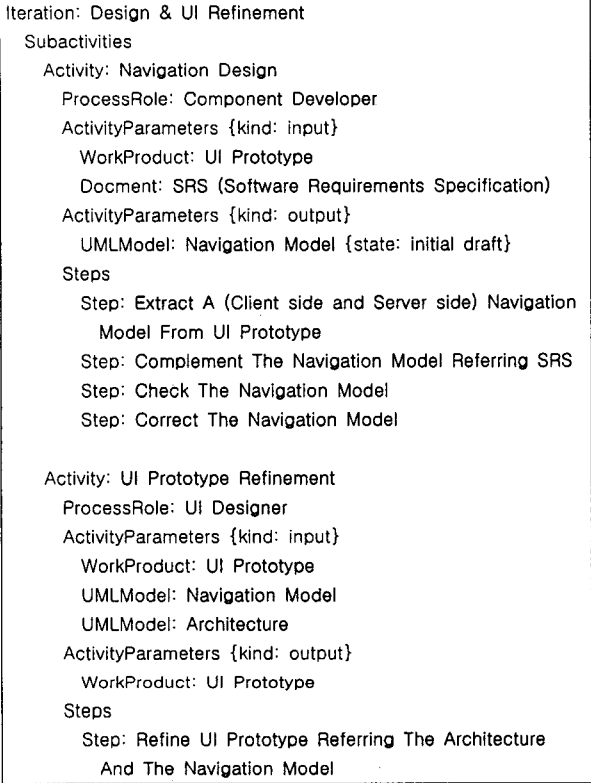


그림 3 항해 설계 활동(Navigation Design Activity)을 포함하는 설계 및 UI 정제 반복(Design and UI Refinement Iteration)의 SPEM 형식 표현(instantiation)의 일부

3.2 UI 프로토타입 정제 활동의 세부 절차

UI 프로토타입 정제 활동은 UI 프로토타입과 항해 모델, 아키텍처를 입력물로 받아들이며 UI 프로토타입에서 틀린 점을 고치고, 부족한 점을 보충하는 작업이다. 따라서 이전의 항해 설계 활동에서 검증 및 수정한 항해 모델을 다시 UI 프로토타입에 반영하는 작업을 포함하고 있다. 활동의 산출물은 정제된 UI 프로토타입이다.

이전의 항해 모델 설계 활동을 거치고, 필요한 경우 항해 설계 활동과 UI 프로토타입 정제 활동으로 이뤄진 소규모 반복을 진행하여 UI 프로토타입을 일정 수준으로 정제할 수 있다.

3.3 컴포넌트 상세 설계 활동에 미치는 영향

항해 설계 활동의 모델 검증 단계를 통해서 컴포넌트 연결 여부를 확인한다. 만약 연결된 컴포넌트가 있다면, 컴포넌트의 인터페이스를 만족하는지 여부를 확인한다. 연결된 컴포넌트가 없는 경우, 새로 개발해야 하는 컴포넌트인데, 항해 모델을 통해서 클라이언트 페이지가 해당 컴포넌트에게 제공할 수 있는 것과 결과 표시 서버 페이지에게 해당 컴포넌트가 돌려주어야 할 것을 정리할 수 있다. 이것은 컴포넌트 개발자의 작업에 도움을 줄 수 있다.

4. 결론

UI 프로토타입은 고객과의 의사소통을 위한 유용한 도구이기 때문에 웹 응용 개발 현장에서 UI 프로토타입을 더욱 중요하게 여기고 있다. 그러나 UI 프로토타입의 복잡성과 비정형성 때문에 항해 설계 활동은 여전히 필요하다. UI 프로토타입에 기반한 항해 설계를 통하여 보다 쉽게 웹 응용 전체를 파악할 수 있고, 사용자 관점 및 서버 관점에서 항해의 문제점을 파악할 수 있다. 그리고 WEE 프로세스의 항해 설계 활동 및 UI 프로토타입 정제 활동의 세부 단계를 밝히고, UML SPEM 프로파일을 사용하여 정리했다.

앞으로 WEE 프로세스의 다른 활동들의 세부 단계를 모두 밝히고, 각 단계 및 활동들의 연관 관계를 정리하여 유연한 프로세스 구성이 가능하도록 할 것이다. 그리고 프로세스 엔진 및 모델링 도구를 개발하여 프로세스 구성 및 설계를 실무적으로 지원할 것이다.

Acknowledgement

본 연구는 한국과학재단 특정기초연구(R01-2002-000-00135-0) 지원으로 수행되었음.

참고 문헌

- [1] D. Schwabe, G. Rossi, L. Esmeraldo, and F. Lyardet, "Engineering Web Applications for Reuse," IEEE Multimedia, Vol. 8, Iss. 1, pp. 2-12, 2001.
- [2] Andreas Kraus and Nora Koch, "A Metamodel for UWE," Technical Report 0301, Ludwig-Maximilians-Universität, München, 2003.
- [3] P. Dolog and M. Bielikova, "Hypermedia Modelling Using UML," In Proc. of ISM, 2002.
- [4] J. Conallen, "Building Web Applications with UML," 2nd ed., Addison Wesley, 2002.
- [5] 이기열, 박상현, 이춘우, 이육진, 정우성, 박영주, 이병정, 김희천, 우치수, "웹 애플리케이션을 위한 UI 중심의 경량 개발 프로세스," 한국 소프트웨어공학 학술대회 논문집, Vol. 1, No. 1, 2005.
- [6] OMG: Software Process Engineering Metamodel, OMG, Ver. 1.1, 2005.
- [7] D. Wallace, I. Raggett, and J. Aufgang, "Extreme Programming for Web Projects," Addison Wesley, 2003.

부록: SPEM 프로파일 (일부)

Stereotype	Base Class	Stereotype Parent	Notation
WorkProduct	Core::Class		
	ActivityGraphs::ObjectFlow		
	State		
Activity	Core::Operation	WorkDefinition	
	ActivityGraphs::ActionState		
	UseCases::UseCase		
ProcessRole	UseCases::Actor	ProcessPerformer	
Document	Core::Class	WorkProduct	
	ActivityGraphs::ObjectFlow		
	State		
UMLModel	Core::Class	WorkProduct	
	ActivityGraphs::ObjectFlow		
	State		