

UI 디자인 프로세스에서의 구조화 수법 연구

Developing Structured System for UI Design Process

이금구, 박영목

국민대학교 테크노디자인전문대학원 인터랙션디자인전공

Lee, Keum-Koo Park, Yeung-Mog

Dept. of Interaction Design,
Graduate school of Techno Design, KMU

• Key words: User Interface Design, UI Design Process, Developing Structured System

1. 서론

UI디자인에서 문제 상황 인식을 위한 구조화 수법은 그 종류가 매우 다양하지만 각 수법들에 대한 표준안이 제시되어있지 않아 정보 전달력이 떨어지고, 각 프로세스별 특징이 고려되지 않고 사용되어 정확한 표현이 이뤄지지 않는 실정이다. 따라서 본 연구에서는 각 구조화 수법들의 특징을 분석하고 UI 디자인 프로세스별 요구사항을 반영하여 각 단계에 적절한 구조화 수법을 적용해보고 그 가능성을 제시하고자 한다.

2. UI 프로세스의 단계 정의 및 요구 사항

UI 디자인 프로세스는 다음과 같이 상황 이해, 컨셉, 디자인, 시뮬레이션, 평가의 네단계로 구분할 수 있다.



[그림 1] UI 디자인 프로세스

먼저 상황의 이해 단계에서는 문제 상황을 인식하기 위해 사용자의 요구분석(Requirement Analysis)이 이뤄진다. 이때 사용자, 대상물, 그리고 상호간의 관계에 대한 충분한 이해가 필요하기 때문에 모든 상황을 자세하게 표현해 주어야 한다. 컨셉 단계에서는 문제 상황에 대한 대안을 제시한다. 발상기법 등을 통한 컨셉 도출을 꾀하기도 한다. 또, 빠른 사고 전개를 해야 하므로 대략적이지만 쉽게 작성할 수 있는 수법이 요구 된다.

디자인 단계에서는 컨셉 단계에서 결정된 대안을 구체화 시킨다. 이 단계 초기에는 아이디어 스케치 같은 저충실도 수법으로 진행되다 결정될 내용이 구조, 조작, 흐름, Feedback, Layout, GUI 등으로 세분화 되면서 점차 고충실도 수법을 사용한다. 즉, 구체적이고 확정적인 형식으로 정리되어야 한다. 시뮬레이션·평가 단계에서는 디자인 단계에서 확정된 내용을 토대로 구현(Prototyping)을 한다. 이 단계에서 다시 컨셉 단계로의 반복이 몇 번 거치면서 Paper Prototyping 같은 간단한 구현에서 점차 실제 대상물에 근접하는 구현이 이뤄져야 한다.

3. 구조화 수법 비교 분석

연구의 대상이 된 다음의 다섯 가지 구조화 수법은 일반적으로 사용되어지거나 기업 내 UI팀에서 사용되어지는 것들로 선별하였다. 먼저 각 수법들의 방법을 관찰하고 동일한 디바이스와 어플리케이션에 각각 적용해 보고 다음과 같은 기준으로 그 특성을 분석하였다.

- 1) 작성 편의 : 해당 수법에 대한 학습, 운용이 편리한가?
 - 2) 표현 능력 : 표현되는 요소와 그에 대한 특성은 어떠한가?
 - 3) 객관성 : 누구나 작성해도 차이가 없는가?
- 분석에 대한 결과는 다음과 같다.

3-1. 순서도 (Flow-Chart)

가장 일반적으로 사용되는 구조화 수법으로 소프트웨어 프로그래밍에서 프로그램 내에서의 처리 경로를 묘사하기 위해 사용되었다. 처리의미를 지닌 특정 기호와 진행방향 표시로 데이터의 흐름을 순서대로 표현하는 방식이다.

작성 편의	-다소 직관적인 표현이기 때문에 작성 소요 시간이 짧다. -문법적인 포맷은 작성자가 그 기준을 정할 수 있어 작성이 쉽다. -모든 과정들이 연결 되어있어 수정이 어렵다. -해당 프로세스의 실행 여부에 대한 판단이 반복된다. -중복되는 프로세스의 생략이 어렵다.
표현 능력	-사용자의 Protocol 데이터와 직접 대응하여 절차상의 문제점을 쉽게 발견할 수 있다. -시간적 흐름에 따라 직선적인 표현이 가능하지만 다른 흐름과의 상호 관계를 나타내기 어렵다. -전체를 나열하는 방식으로 표현되어 전체 메뉴의 깊이(Depth)를 표현해주기 어렵다 -순서도에 사용되는 기호(세이프)의 의미를 숙지하고 있어야 작성 및 분석이 가능하다.
객관성	-주관성이 개입될 여지가 있어 일관성과 표현 방법에 대한 기준이 모호하다.
기타	-절차 지향

3-2. 화면구조 (Screen Flow-Chart)

버튼이나 특정 액션에 의한 디스플레이를 순서대로 나열하여 표현한다. 일반적으로 웹 디자인에서 많이 사용되어진다.

작성 편의	-화면의 디자인이나 배치, 버튼 등이 결정되어야 하기 때문에 어느 정도 결정된 대상이나 이미 존재하는 대상이어야 한다. -이미 반복된 과정이더라도 모두 표시되어야 한다.
표현 능력	-디스플레이나 화면의 레이아웃이 직접적으로 표현되어있어 이해하기가 쉽다. -사용자의 Protocol 데이터와 직접 대응하여 절차상의 문제점을 쉽게 발견할 수 있다. -시간의 흐름과 기능간의 이동경로가 중, 횡의 레이아웃으로 자연스럽게 표현된다. -모두 과정들이 연결 되었으므로 수정이 용의하지는 않다. -버튼의 기능에 1:1대응하므로 설계 및 디자인에 용이하다.
객관성	-디스플레이 이미지와 버튼 이미지만으로 표현하기 때문에 객관적 표현이 가능하다.
기타	-절차 지향

3-3. OCD (Operation and Control Diagram)

A사에서 제품디자인의 기능을 표현하기 위해 사용되는 수법으로, 버튼을 중심으로 진행되는 액션을 기술한다. OCS (Operation and Control Scrip)를 병행하여 상황에 대한 묘사를 보충하기도 한다.

작성 편의	-학습이 쉽고 설계 이후 수정이 쉽다. -속성이 계승되어 반복적 표현을 피할 수 있다. (Memory header) -버튼 중심으로 이루어져 다양한 동작표현이 어렵다. (abstract operation라는 기능이 있지만 모든 동작을 표현하기엔 한계가 있다.)
표현 능력	-버튼에 의한 기능 흐름을 잘 나타낼 수 있다. -입장, 퇴장, 종료의 관계표현이 명확하다. -세부 프로세스를 표현하기 쉽다. -목표 지향이므로 구성요소들 간의 관계를 표현하기가 어렵다. -전체 프로세스 상에서의 레벨을 표현해 주기가 어렵다. -횡적 관계가 많은 Application을 표현하기 어렵다.
객관성	-기능의 흐름에 대한 객관성이 보장된다.
기타	-절차 지향 -OSC(입장동작, 퇴장동작 기술)병행

3-4. Usability Matrics

B사에서 사용되는 태스크 구조 분석 방법으로 하나의 태스크에 대한 유사성을 기준으로 다른 태스크들을 관련짓는다. 전문가가 결정하지 못하는 모호한 부분은 설문이나 카드 소팅으로 그 유사성을 결정한다.

작성 편의	-직관적인 기능 분류로 표현방식이 간단하다.
표현 능력	-매트릭스 구조로 표현되어 전체 구조를 파악하기 어렵다.
객관성	-직관에 의한 분류이기 때문에 작업자나 참여 그룹에 대한 신뢰도를 확보하지 못할 때 객관성이 결여될 수 있다.
기타	-절차 지향 -설문과 카드 소팅을 통해 사용자가 직접 참여할 수 있다.

3-5. OMT (Object Oriented Model Technique)

소프트웨어 공학에서 개발된 수법인 객체지향 수법을 토대로 구조모델링과 동적모델링으로 표현한다. 객체지향 설계에 대한 충분한 이해가 무엇보다 중요하다. 특히 기존의 절차지향의 구조화 수법과 혼동하여 그 개념이 모호해 지거나, 각 객체의 정확한 정의의 부재로 인하여 객체간의 관계가 모호해 지는 경우가 많다.

작성 편의	-직관적인 기능 분류로 표현방식이 간단하다. -설계 기법의 학습이 어렵다. -객체화 되어있으나 설계 이후에 분류 기준이 수정되면 전체적 구조에 대한 수정이 불가피하다. -기존의 설계 기법으로 작성된 구조도와와의 호환이 어렵다.
표현 능력	-객체의 속성에 대한 판단으로 실세계의 현상을 보다 정확히 모델링 할 수 있다. -구성 요소들 간의 관계를 명확하게 표현할 수 있다. -시간적 요소의 표현이 가능하다.(동적 모델) -모델링 된 이후에 전체 구조를 이해하기가 쉽다.
객관성	-객체에 대한 명확한 정의를 전제로 하기 때문에 객관성이 보장된다.
기타	-객체 지향

4. 결론

앞서 언급한 각 수법들에 대한 장단점을 토대로 프로세스 상의 각 단계별 요구사항에 적절한 수법을 다음과 같이 선정해 보았다.

상황 이해 단계	
수행 작업	- 사용자 분석 : 사용자 작업모델 (작업 전, 작업 중) - 제품 분석 : 화면 구조, 기능 구조, 기계 구조 - 사용자-제품 분석 : 비교 분석
결과	- 대상의 문제점
적용 수법	- OMT : 요소들의 관계 파악, 모든 상황 표현

컨셉 단계	
수행 작업	- 문제 해결을 위한 대안 제시 - 기능별 카테고리 결정
결과	- 조작 태스크 - 스타일 가이드
적용 수법	- 발상기법 - Usability Matrics : 태스크 분류 - 순서도 : 신속하고 직관적 표현

디자인 단계	
수행 작업	- 구체적 조작 방법 결정 - 레이아웃 구성 - 세부 피드백 결정 : 디스플레이 액션, 사운드 등
결과	- 세부 조작 태스크 - 레이아웃
적용 수법	- OMT : 모든 요소들에 대한 정의 - 화면 구조도 : 디스플레이의 구현 - OCD : 세부 기능 표현

시뮬레이션·평가 단계	
수행 작업	- 구체화 - 평가
결과	- 시각화 결과 - 수정, 보완 사항
적용 수법	- 기타 프로그래밍 언어

사실 이처럼 단계마다 각 구조화 수법을 모두 표현하기에는 효율적 측면에서 무리가 따를 수 있을 것이다. 때문에 향후, 위 사항을 토대로 하여 프로세스별 요구사항을 적절히 수용하는 수법을 제작, 개선하고, 전체 프로세스 상에서 각 수법들에 대해 일관된 정보흐름을 유지할 수 있는 관리체계를 갖추게 된다면 UI 디자이너에게 있어 보다 체계적이고 효율적인 구조화 툴을 확립할 수 있을 것으로 사료된다.

참고문헌

- The Usability Engineering Lifecycle : Deborah J. Mayhew 1999
- 사용자 인터페이스 개발 Process 확립 및 Tool의 개발 : 산업 자원부 보고서 (주) 우퍼 디자인