

# UML 2.0 행위 다이어그램을 확장한 비동기 상호작용 웹 애플리케이션의 항해 모델

변창현<sup>○</sup>, 서정호, 류성열  
 송실대학교 컴퓨터학과

chams<sup>○</sup>@ssu.ac.kr, hoya6246@ssu.ac.kr, syrheo@ssu.ac.kr

## A Navigation Model of Asynchronous Interaction Web Applications with Extended Behavioral Diagrams of UML 2.0

ChangHyun Byun<sup>○</sup>, JungHo Seo, SungYul Rhew  
 Dept. of Computer, Soong-Sil University

### 요 약

Web2.0의 비동기식 웹 페이지 전환기술이 주목을 받으면서 기존의 웹사이트들이 동적인 웹 사이트로 바뀌어 가고 있다. 그러나 웹 애플리케이션 개발에 적용 가능한 설계모델인 UML2.0은 정적인 웹 페이지 설계 모델로 국한되어 있다. 따라서 본 논문에서는 UWE의 항해모델에서 뷰 관점 항해모델과, 데이터 전송 항해 모델을 State Machine, Communication Diagram을 이용하여 비동기식 웹 페이지 항해 모델 개선과 기존의 동기식 웹 페이지 모델과의 호환성 방안을 제안한다.

[1].

### 1. 서 론

지난 2004년도에 개최한 1st Web 2.0 Conference에서 Dale Dougherty가 처음으로 언급한 Web 2.0이란 키워드가 인터넷 산업 분야 전반에 걸쳐 많은 사회적 변화를 양상하고 있다. 특히 많은 웹사이트들이 Google사의 Google Docs & Spreadsheets라는 애플리케이션과 같이 네트워크 부하가 상대적으로 적은 비동기식 웹 페이지 전환으로 바뀌어 나가고 있으며, 이로 인해 웹상에서도 데스크탑 수준의 애플리케이션 구현이 가능하게 됐다.

그러나 기존 웹사이트들이 동적인 웹 애플리케이션으로 변화하고 있는 반면에, 웹 애플리케이션 개발에 적용 가능한 설계모델들은 기존의 정적인 웹 페이지 설계 모델에만 국한되어 있는 실정이다. 이를 개선하기 위해 본 논문에서는 UML 2.0을 확장한 항해 모델에서 뷰 관점 항해모델과, 데이터 전송 항해 모델을 State Machine, Communication Diagram을 이용하여 비동기식 웹 페이지 항해 모델 개선과 기존의 동기식 웹 페이지 모델과의 호환성 방안을 제시한다.

### 2. 관련연구

#### 2.1 비동기 상호작용: Ajax

Ajax(Asynchronous JavaScript + XML)는 Lazlo, JSF, Flex와 같이 RIA(Rich Internet Application) 지향의 웹 애플리케이션의 개발 요구사항을 충족시켜준다. 웹상에서 데스크탑 수준의 애플리케이션 구현을 위한 Ajax에서의 주요 특징은 아래 [표 1]과 같이 3가지를 들 수 있다

[표 1] Ajax의 주요 특징

소규모 서버 측 이벤트	<ul style="list-style-type: none"> <li>서버에 소규모 데이터만 요청하여 받아옴</li> <li>DOM 트리 조작으로 페이지 모양 변경</li> <li>전체 웹 페이지의 Refresh가 없음</li> </ul>
비동기 방식 통신	<ul style="list-style-type: none"> <li>서버로 요청을 보내는 것이 브라우저를 멈추게 하지 않음</li> <li>요청을 처리하는 시간 동안 애플리케이션의 다른 작업이 가능</li> </ul>
데스크탑 수준의 이벤트	<ul style="list-style-type: none"> <li>데스크탑 애플리케이션 수준의 사용자 이벤트와 동등한 수준(마우스 클릭, 마우스 오버, 드래그 등)</li> <li>어떠한 사용자 이벤트도 비동기 요청을 만들 수 있음</li> </ul>

웹 페이지를 항해 시, 사용자와의 상호작용이 어려워지는 중요한 원인으로는 네트워크 지연 현상을 꼽을 수 있다. 기존의 웹 페이지는 클라이언트에서 사용자 작업(이벤트)의 처리 요청 시, 동기식 통신 방식의 서버로부터 응답이 되돌아올 때까지 사용자는 항상 대기 상태로 있을 수 밖에 없다는 것과, 처리 요청에 따라 네트워크 누적 전송량이 계속해서 증가한다는 단점이 존재한다[2].

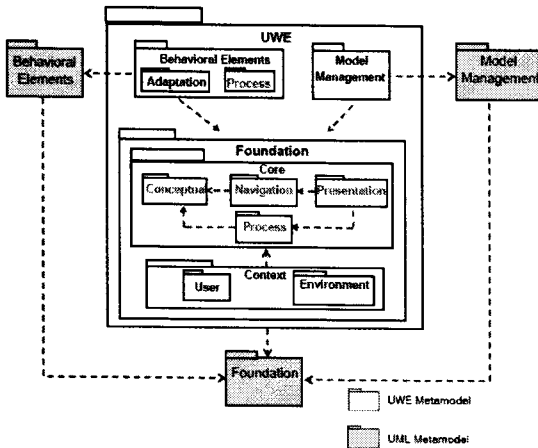
그러나 Ajax를 이용하여 웹 애플리케이션을 개발 했을 시, XMLHttpRequest 고유 객체에 의해 비동기 방식 통신으로서 데이터의 처리가 가능해지기 때문에 페이지 Refresh가 없어, 클라이언트에서 서버로의 다수의 이벤

트 처리 요청 시에도 사용자는 다른 작업이 가능해지고 네트워크 전송량이 거의 일정하게 유지될 수 있다[3].

2.2 UWE메타 모델의 패키지 형태

UML specification 문서에는 각 다이어그램 메타 모델이 제시되어 있다. 메타 모델은 다이어그램 단위 별로 모델을 기술할 때 그 표기법의 이해를 높이고, 설계 작업에 있어서 정확한 표현이 되도록 도와준다.

UML의 확장은 메타 모델을 통해 필요한 형태로 확장이 가능하다. [그림 1]에서는 UML을 기반으로 하고 있는 UWE의 메타모델 패키지 구조를 보여주고 있다[4].



[그림 1] UWE 메타모델의 패키지 구조

2.3 기존의 항해 모델의 형태

웹 애플리케이션 개발에 적합하도록 확장된 모델링 언어 다이어그램은 [그림 2]와 같이 Conceptual Diagram, Navigation Diagram, Presentation Diagram이 존재한다 [5].



[그림 2]. UWE 설계 프로세스 모델

이중 항해 모델(Navigation Diagram)은 웹 페이지 항해에 있어서 CCN(클라이언트 페이지에서 클라이언트 페이지로의 항해), CSN(클라이언트 페이지에서 서버 페이지로의 항해), SCN(서버페이지에서 클라이언트 페이지로의 항해), SSN(서버 페이지에서 서버 페이지로의 항해)과 같이 크게 네 가지 분류로 나눌 수 있다.

CCN은 데이터 전송이 존재하지 않는 단순 링크 형태를 말하며 CCN을 제외한 나머지 세 가지 경우는 예를 들어 로그인 페이지와 같이 서버에서 관리되는 사용자의 세션 정보가 오고 가는 페이지의 항해 형태를 말한다[6].

2.4 비동기식 항해 모델을 위한 확장된 메타모델

이기열이 제안한 웹 애플리케이션의 항해모델은 State

Machine Diagram을 확장한 메타모델을 제안하였다. 이기열의 연구에서 제안한 [그림 3]의 Transition, State 클래스는 OMG에서 제공하는 UML Superstructure의 BehaviorStateMachine 패키지에 속하는 클래스이다[7]. 다음 하위 클래스인 Navigation, Submit, PageView, VirtualPageView는 항해 모델을 위한 확장된 클래스로서 웹 애플리케이션에 적합한 항해 모델을 제시하고 있다.

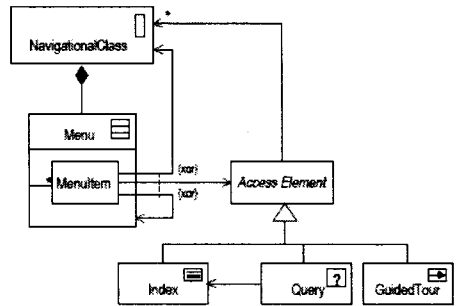


[그림 3] 항해 모델을 위한 UML 상태 기계 다이어그램의 확장

UML을 확장하여 웹 애플리케이션에 적합한 항해 모델을 위해 이기열의 연구에서는 데이터 전송 관점의 항해 모델은 Sequence Diagram을 사용하였다. 그러나 Sequence Diagram은 비동기식 흐름을 표현하기에는 급격하게 복잡해지는 단점을 가지고 있다[6].

2.5 UWE 항해 모델링 요소들의 메타모델

[그림 4]는 UWE에서 제안한 항해 모델링 요소들에 대한 메타모델이다. "NavigationalClass"에서 "Access Element" 하위 계층에 항해 모델에 필요한 모델링 요소들을 정의하고 있다.



[그림 4] UWE 항해 모델링 요소들의 메타모델

3. 비동기식 항해 모델을 지원하기 위한 UML의 확장 표기법

항해 모델 확장을 위해 메타모델을 작성하는 것은 새로운 연결들의 추가에 의해 복잡해질 수 있으나, 한편으로는 추가적인 연결들에 의해 더 높은 항해의 가독성을 향상시킬 수 있으며 더욱 체계적인 설계에 도움을 주게 된다[5]. 따라서 본 논문에서는 비동기식 상호작용을 지원하는 확장된 뷰/데이터 전송 관점 항해 모델에서 State Machine Diagram과 Communication Diagram에서 쓰일 항해 경로 타입 표기법과 항해 노트 타입 표기법을 제시한다.

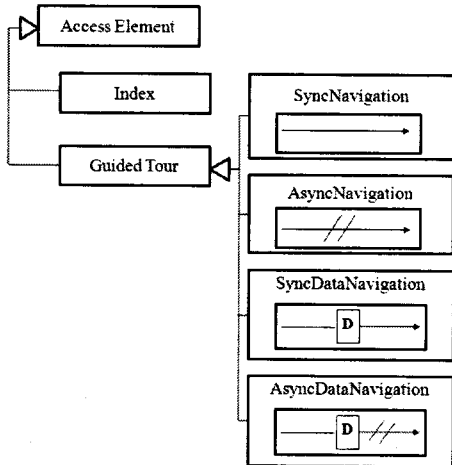
3.1 확장된 항해 경로 타입의 표기법

[표 2]에서와 같이 동기 및 비동기 상호작용을 표현하는 항해 경로 타입을 정의하였다. SyncNavigation과 AsyncNavigation은 일반적인 뷰 관점에서의 항해 경로 타입으로 정의하고, SyncDataNavigation과 AsyncDataNavigation에서는 데이터 전송 관점에서의 항해 경로 타입으로서 정의한다.

[표 2] 확장된 항해 경로 타입의 표기법

경로타입	표기법
SyncNavigation	
AsyncNavigation	
SyncDataNavigation	
AsyncDataNavigation	

[그림 4]는 UWE에서 제안한 항해 모델링 요소들의 메타 모델을 확장한 형태이다. 본 연구에서 데이터 흐름에 관한 항해 경로 타입을 정의했기 때문에, UWE에서 제안한 “Query” 모델링 요소는 제외하였고 “GuidedTour” 모델링 요소에 일반적인 연결관계와 데이터 전송 연결관계를 포함하였다.



[그림 4] 확장된 항해 모델의 메타 모델의 공통 경로타입 요소

사용할 State Machine Diagram, Communication Diagram의 공통적으로 쓰일 노트 타입의 표기법을 일반적으로 사용자에게 보여지는 페이지와(General Page) 로그인 페이지와 같은 데이터 전송이 행해지는 페이지(Data Page), 비동기식 웹 애플리케이션의 링크 경우(Async Page)로 정의하였다.

AsyncPage의 경우 데이터 전송이 비동기로 행해지기 때문에 별도로 데이터 전송의 노트 타입을 설정하지는 않았다.

[표 3] 확장된 항해 노트 타입의 표기법

노트타입	표기법
General Page	
Data Page	
Async Page	

4. 비동기 상호작용 지원 항해 모델

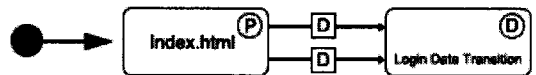
4.1 뷰 관점 항해 모델의 사례

본 연구의 항해 모델을 증명하기 위해 [표 4]의 간단한 온라인 도서관 시스템 시나리오를 예로 하였다.

[표 4] 온라인 도서관 시나리오

S1	사용자는 온라인 도서관 웹 사이트에 접속한다.
S2	사용자는 사용자 계정 페이지에서 로그인 또는 로그아웃을 한다.
S3	사용자는 도서를 검색, 예약, 예약 취소 작업을 요청하는 동안 다른 작업 수행이 가능하다.

[그림 5]는 온라인 도서관 시스템에서 로그인 시나리오를 예로 하여 동기식 상호작용의 항해 모델을 표현하였다.



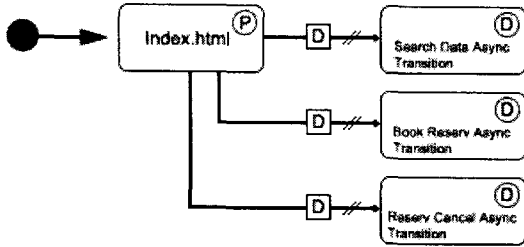
[그림 5] 온라인 도서관 시스템 뷰 관점의 동기식 항해 모델의 예

[그림 6]은 State Machine Diagram을 확장하여 비동기식 상호작용을 지원하는 항해 모델의 예이다. 온라인 도서관 시스템 시나리오를 토대로 비동기식 상호작용의 항

3.2 확장된 항해 노트 타입의 표기법

[표 3]에서는 뷰/데이터 전송 관점에서의 항해 모델로

해 모델을 표현하였다.

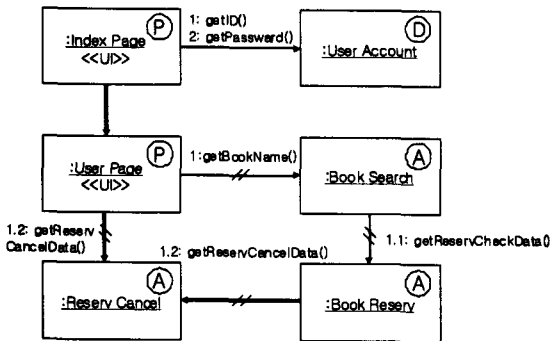


[그림 6] 온라인 도서관 시스템 뷰 관점의 비동기식 항해 모델

#### 4.2 데이터 전송 관점 항해 모델의 사례

하이퍼텍스트의 특성상 비순차적인 페이지간의 항해 흐름을 고려하여 동기 및 비동기식 상호작용이 적용된 페이지간의 연결 관계를 강조하고자 Communication Diagram을 사용하였다.

앞 절에서와 같은 시나리오를 바탕으로 하여 [그림 7]에서는 Communication Diagram을 확장하여 비동기 상호작용을 지원하는 데이터 전송 관점 항해 모델을 표현하였다. 공통된 경로 타입과 노드 타입을 사용하여 항해 모델로서의 가독성을 높이고 각각의 항해 경로에 따른 데이터 전송의 관련 함수를 표시하였다.



[그림 7] 비동기식 온라인 도서관 시스템 데이터 전송 관점 항해 모델의 예

### 5. 결론 및 향후 방향

본 연구에서는 비동기식 상호작용을 지원하는 항해 모델을 뷰 관점과 데이터 전송 관점으로 제안하였다. 각각의 관점에 따른 항해 모델은 UML 2.0 표준에서 정의한 State Machine Diagram과 Communication Diagram을 확장하여 새로운 표현법으로 표현하였으며, 도서관 시스템을 예로 하여 비동기식 항해 모델을 표현할 수 있음을 보였다.

본 논문에서 새롭게 제안한 비동기식 항해모델 표현법은 정적인 웹 페이지 설계 모델에만 국한되어 있는 기존의 표현법 보다 비동기식 웹 페이지 항해 모델 표현에

적합하며, 동기식 웹 페이지 항해 모델표현에도 호환되어 사용 가능하다.

향후 연구 방향으로서는 비순차적인 하이퍼링크의 특성을 더욱 명확하게 명시해 줄 수 있는 메타모델의 확장이 필요할 것이며, 비동기식 상호작용에 있어서 서버와 클라이언트 간의 실시간 데이터 요청/응답 사이에 작업 처리에 대한 우선순위를 표현해주는 방안이 고려되어야 할 것이다.

#### 참고문헌

- [1]Gehntland, Ben Galbraith, Dion Almea, "Pragmatic Ajax", 2006
- [2]Eric Pascarello, Dave Crane, "Ajax in Action", Manning Publications, 2005.
- [3]Jesse James Garrett, "Ajax: A New Approach to Web Application", 2005
- [4]Nora Koch, "UWE: UML-based Web Engineering" at the Universidad de Sevilla, Spain, tutorial, 21.04.2005
- [5]Rolf Hennicker, Nora Koch, "Modeling the User Interface Web Applications with UML", Ludwig-Maximilians-University Munich, Germany
- [6]이기열 외5인, "확장된 UML 모델을 이용한 기민한 웹 애플리케이션 방법론", 정보과학회논문지 소프트웨어 및 응용 제34권 제3호, 2007.03
- [7]OMG, "Unified Modeling Language: Superstructure 2.1.1", OMG UML Specification, 02. 2007