

HTML5 기반 컴포넌트에 대한 코드 어시스트 구현 방안 연구

강재규*, 유연희*, 최용호*, 김장군*, 박현범*

*(주)유코아시스템 부설연구소

e-mail:jgkang@u-core.co.kr

A Study on Code Assist Method of HTML5 Components

Jea-Gyu Kang*, Yeon-Hee Yu*, Yong-Ho Choi*, Jang-Goon Kim*, Hun-Beom Park*

*Research Institute, Ubiquitous Core System

요 약

표준 웹 환경의 확산으로 HTML5를 기반으로 웹 저작을 지원하는 많은 WYSIWYG HTML5 저작도구들이 등장하고 있다. 그러나 대부분의 WYSIWYG HTML5 저작도구는 컴포넌트 이벤트에 대한 기본적인 함수 작성 기능만 제공 할 뿐 컴포넌트 제어를 위한 코드 어시스트 기능을 제공하지 않고 있다. 본 논문에서는 이클립스 JSDT(JavaScript Development Tools) 코드 어시스트개념에 대해 기술하고, 이를 기반으로 WYSIWYG HTML5 저작도구에서 제공하는 컴포넌트 제어를 위해 코드 어시스트 기능을 구현하였다. 이로 인하여 WYSIWYG HTML5 저작도구에서 제공하는 컴포넌트에 대한 사용자들의 접근성과 개발 생산성 증대가 기대된다.

1. 서론

현재 대부분의 IDE(Microsoft Visual Studio, Eclipse, Delphi 등)들에서는 코드 어시스트 기능이 필수 기능으로 자리 잡고 있다. 제공 되는 코드 어시스트 기능의 수준에 따라 해당 저작도구에 대한 선택이 좌우되고 있는 상황이다. 그만큼 코드 어시스트의 기능이 개발 생산성에 직접적인 기여를 하고 있는 것이다. 그런데 대부분의 WYSIWYG HTML5 저작도구에서는 기본적인 자바스크립트 문법 코드 어시스트 기능은 제공 하고 있지만 컴포넌트 제어를 위한 자바스크립트 코드 어시스트 기능을 제공하지 않고 있다. 이로 인하여 새로운 컴포넌트에 대한 사용자들의 접근과 생산성을 저하시키고 있다.

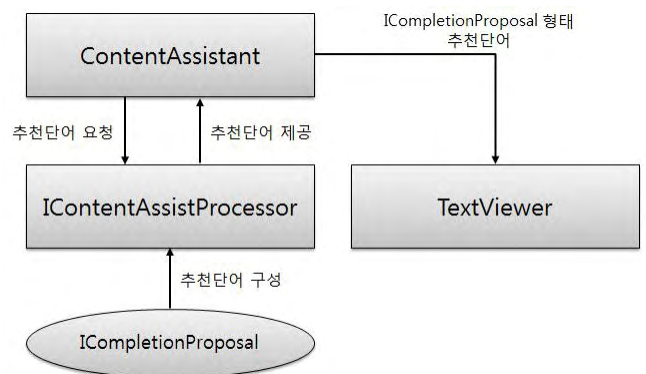
이렇게 WYSIWYG HTML5 저작도구 자바스크립트 에디터의 코드 어시스트 기능이 제한적으로 제공되는 이유는 자바스크립트가 기본적으로 데이터 형에 대한 문법을 제공하지 않기 때문이다. 그렇기 때문에 자바스크립트에서는 함수의 매개변수로 전달된 변수의 데이터 형을 추적하기 힘들고 이로 인해 에디터에서 전달된 데이터에 대한 정확한 코드 어시스트를 제공할 수 없는 것이다.

본 논문에서는 저작도구 컴포넌트에 적합한 코드 어시스트 기능을 제공하기 위하여 JSDT(JavaScript Development Tools) 코드 어시스트 기능의 분석을 기술한다. 이를 기반으로 컴포넌트 코드 어시스트 기능을 설계/구현 하였다.

2. 이클립스 코드 어시스트

2.1 이클립스 코드 어시스트 개념

이클립스에서는 코드 어시스트 기능을 위하여 에디터와 인터페이스, 클래스들을 제공하고 있다. ContentAssistant, IContentAssitProcessor, ICompletionProposal 등이 이클립스에서 코드 어시스트를 위하여 제공하는 기본적인 클래스 및 인터페이스들이다.

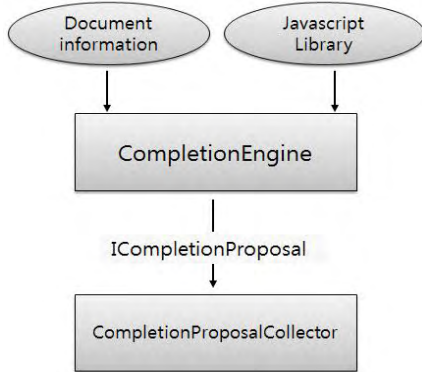


(그림 1) 이클립스 기본 코드 어시스트 기능도

이클립스는 ContentAssistant를 통하여 코드 어시스트 기능을 제공하는데 ContentAssistant가 TextViewer를 참조하고 있으며 사용자가 추천단어를 요청할 때 ContentAssistant의 함수를 호출하여 IContentAssitProcessor에서 추천 단어를 ICompletionProposal의 인스턴스로 구성하여 제공한다.

2.2 이클립스 JSDT 코드 어시스트 개념

JSDT의 코드 어시스트 개념도 이클립스의 기본 코드 어시스트개념에 대한 확장이라고 할 수 있다. JSDT는 이클립스의 기본 어시스트 개념에 자바스크립트 라이브러리와 Document 스캔 기능이 추가 되고 이들을 이용하여 사용자에게 적합한 추천단어의 생성 및 분류를 담당하는 엔진 기능도 같이 추가 되었다.



(그림 2) JSDT 코드 어시스트 기능도

사용자가 어시스트 기능을 호출하면 자바스크립트 에디터는 에디터에 등록된 ContentAssistant의 함수를 호출하여 CompletionEngine가 생성한 추천단어를 얻어 온다. CompletionEngine은 현재 수정중인 Document의 전체 내용과 커서의 위치를 제공 받아 코드 어시스트의 키워드를 알아내고 해당 키워드와 관련된 라이브러리를 찾아 사용자에게 라이브러리의 내용을 추천단어 형태로 가공하여 제공한다.

JSDT에서는 추천단어의 정보를 제공하는 ICompletionProposal 인터페이스가 확장이 되는데 이는 추천단어에 매개변수를 갖는 함수에 대한 매개변수 가이드 기능과 추천단어를 전시할 때에 사용자가 입력한 키를 통해 추천단어를 필터링하는 기능 등이 추가되기 때문이다.

<표 1> JSDT 코드 어시스트 추천단어 함수 호출 스택

Stack 방향	함수 [클래스]
↑	getJavaCompletionProposal() [CompletionProposalCollector]
	...
	computeCompletionProposals() [JavaCompletionProcessor]
	...
	showPossibleCompletions() [ContentAssistant]

JSDT에서 기본 이클립스의 코드 어시스트의 기능에 대하여 많은 부분을 확장하여 사용자에게 기능을 제공하지만 기본적으로는 이클립스의 기본 코드 어시스트 기능

메커니즘을 따른다. 이는 JSDT 코드 어시스트의 기능에 대한 수정 필요성이 있을 경우 JSDT 코드 어시스트를 분석하여 이클립스 기본 코드 어시스트 기능 부분을 찾아내고 해당 부분에서 기능을 수정할 수 있다는 것을 의미한다. <표 1>에서 보여주듯이 JSDT에서는 ContentAssistant 클래스에게 추천 단어를 제공해주는 JavaCompletionProcessor(IContentAssistProcessor 구현 클래스) 클래스의 computeCompletionProposals 함수에서 CompletionEngine에서 전달 받은 추천단어를 최종적으로 수집하고 추천 단어 전시를 담당하는 ContentAssist에 전달한다.

3. 저작도구 컴포넌트용 코드 어시스트 구현

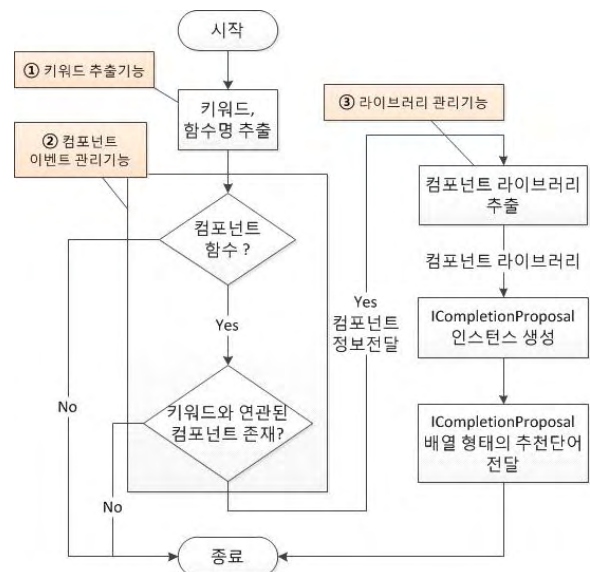
현재 자사에서서는 HTML5 기반의 WYSIWYG 저작도구(HOP, HTML5 Open Platform)를 이클립스 플러그인으로 개발 중에 있으며 저작도구에서 제공하는 컴포넌트에 대한 코드 어시스트 기능을 본 논문의 연구로 구현하였다.

3.1 컴포넌트 코드 어시스트 설계

컴포넌트 코드 어시스트 기능을 구현하기 위해 본 논문에서 구현하는 컴포넌트 코드 어시스트 기능은 다음과 같이 구성하였다.

- 라이브러리 관리기능 : 컴포넌트의 라이브러리를 로드 및 관리
- 컴포넌트 이벤트 관리 기능 : 컴포넌트의 이벤트 함수를 등록하고 관리
- 키워드 추출 기능 : 사용자가 입력한 키워드를 Document에서 추출

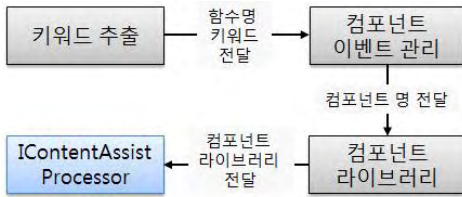
위 기능들은 IContentAssistProcessor 구현 클래스의 computeCompletionProposals 함수에서 수행되며 컴포넌트 코드 어시스트 기능의 흐름은 (그림 3)과 같다.



(그림 3) 컴포넌트 코드 어시스트 기능 흐름도

3.2 컴포넌트 코드 어시스트구현

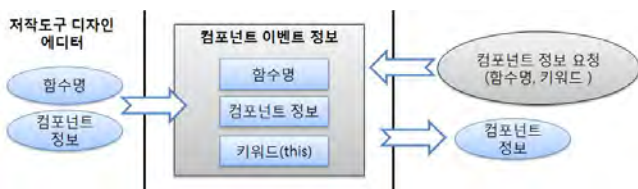
컴포넌트 어시스트 기능의 구현은 3.1절의 각 기능들에 대한 구현과 ICompletionProposal의 확장으로 수행된다. 구현 기능들의 구성은 (그림 4)와 같다.



(그림 4) 컴포넌트 어시스트 기능 구성도

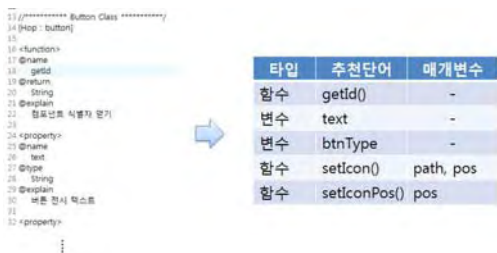
키워드 추출 기능은 사용자가 코드 어시스트 기능을 실행시킬 때 현재 Document에서 커서의 위치를 이용하여 코드 어시스트를 실행시키는 키워드를 찾는다. 또한 해당 커서가 위치한 곳의 함수명도 찾는다. 키워드는 현재 커서에서 이전 문자들을 수집하여 전달하고 함수명은 문자들을 수집하여 함수 형태의 문자열 패턴을 찾아 함수명만 추출하여 전달한다.

컴포넌트 이벤트 관리기능은 저작도구의 컴포넌트 디자인 에디터에서 컴포넌트 이벤트 등록 시 이벤트 함수명과 컴포넌트의 정보를 전달 받는다. 여기에 컴포넌트 대표 키워드를 기본적으로 같이 저장한다. 컴포넌트정보를 요청 받을 때 제공 받은 함수명과 키워드를 이용하여 컴포넌트 정보를 조회하고 일치하는 컴포넌트를 제공해준다.



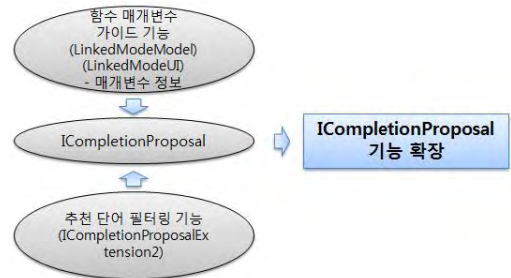
(그림 5) 컴포넌트 이벤트 관리기능

컴포넌트 라이브러리 기능은 컴포넌트의 멤버 함수 및 변수에 대한 정보를 저장하고 특정 컴포넌트에 대한 정보를 요청 받을 경우 보유하고 있는 라이브러리 정보를 확인하고 해당 컴포넌트에 속하는 라이브러리 정보를 추출하여 전달한다. 라이브러리 텍스트 정보는 사용자가 손쉽게 수정/저장 할 수 있는 텍스트 파일 형식으로 구성한다.



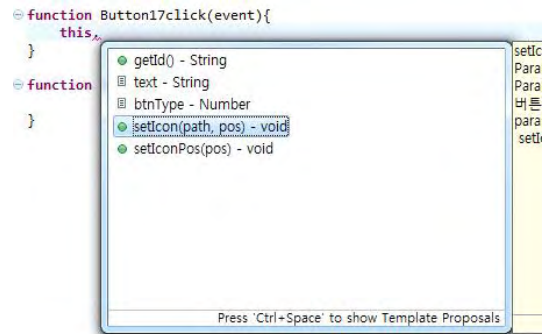
(그림 6) 컴포넌트 라이브러리 로드

추천 단어의 정보를 담아 전달되는 ICompletionProposal의 인스턴스도 확장 된다. 확장된 내용은 (그림 7)과 같다.



(그림 7) ICompletionProposal 기능확장

컴포넌트 어시스트의 기능에 대한 구현은 (그림 8)과 같다. (그림 6)에서 로드한 컴포넌트 라이브러리를 어시스트의 추천단어로 제공하고 있다.



(그림 8) 컴포넌트 코드 어시스트 구현

4. 결론

본 논문에서는 이클립스 JSDT 코드 어시스트개념에 대해 기술하였고, 이를 기반으로 WYSIWYG HTML5 저작도구에서 제공하는 컴포넌트 제어를 위해 코드 어시스트 기능을 구현을 기술하였다. 이로 인하여 본 논문에서 구현한 컴포넌트 코드 어시스트 기능이 적용된 WYSIWYG HTML5 저작도구의 컴포넌트에 대한 사용자들의 접근성과 개발 생산성 증대가 기대된다.

향후에는 사용자에게 보다 편리한 컴포넌트 코드 어시스트 기능을 제공하기 위한 컴포넌트 선택자(원하는 컴포넌트를 찾아서 변수에 할당) 기능을 확장할 예정이다.

본 논문은 2013년부터 2015 년도까지 총 2년 동안 기술혁신 개발 사업으로 중소기업청의 지원을 받아 연구가 진행 중에 있음 (과제번호: S2081915)

참고문헌

- [1] SWT/JFace 인 액션(에이콘)
- [2] Building an Eclipse Text Editor with JFace Text (www.realsolve.co.uk/site/tech/jface-text.php)
- [3] JavaScript Development Tools (www.eclipse.org/webtools/jsdt/)