

# HTML5를 활용한 온라인 지도 기반 AIS선박 모니터링 시스템 구현

† 이서정 · 이재욱\*

†, \* 한국해양대학교 IT공학부

## A Development of AIS Vessel Monitoring System on online map using HTML5

† Seo-Jeong Lee · Jae-Wook Lee\*

†, \* Division of IT Engineering, Korea Maritime University, Busan 606-791, Korea

**요 약** : 최근 선박의 안전운항에 대한 요구가 증가되면서, 국제해사기구에서는 SOLAS규정에 따라 선박의 다양한 정보를 제공하는 AIS의 의무 장착을 시행하고 있다. AIS장비는 선박의 자선 정보를 주변 선박 및 육상에 자동으로 브로드캐스팅방식으로 발신하고, 또한 주변 선박의 정보를 수신할 수 있는 장비이다. 최근에는 다양한 디스플레이 장치에 서비스를 제공하기 위한 시도가 진행되고 있으며, 특히 웹 응용을 이용하는 사례가 많아지고 있다. 본 논문에서는 이러한 웹 기반 응용에 대한 실현을 실험해보기 위해, 온라인 기반의 상용지도를 이용해 AIS 정보를 확인할 수 있는 디스플레이 시스템을 구현해 보았다. AIS수신 메시지 분석을 통해 주변 선박 정보를 분류하고, 상용지도는 구글 맵을 사용했으며, 디스플레이 처리를 위해서는 차세대 웹 표준인 HTML5 기술을 이용하였다.

**핵심용어** : 선박자동식별장치, 웹기반 응용, 웹개발 플랫폼, HTML5, Ajax, 웹소켓

**Abstract** : As the increasing of requirement for safety navigation, IMO has enforced the mandatory installation of vessel AIS equipment by SOLAS regulation. The AIS transceiver broadcasts various vessel information which can be gathered by the receivers on-board or on-shore. And, recently, as web-based application developments on various devices have been increased, there are more and more requirements of AIS information presentation on internet. To meet these web-based application requirements, this paper shows the practical implementation of the AIS display system, which is on the Google maps as online commercial map and adopts the HTML5 as a web standard.

**Key words** : AIS, Web-based application, Web development platform, HTML5, Ajax, Websocket

### 1. 서 론

국제 교통량이 증가에 따라 자연히 해상 교통량도 같이 증가하고 있다. 이러한 해상 교통량 증가는 해상사고의 위험을 증가 시키는 주요 요인이 되고 있다(박 등, 2009). 항해안전정보 제공의 일환으로, IMO SOLAS( International Convention for the Safety of Life At Sea, 해상인명안전협약) 규정에 따라, AIS( Automatic Identification System, 선박자동식별장치)의 의무 장착을 시행하고 있다. 대상선박은 모든 여객선 및 국제항해에 종사하는 300톤 이상의 선박과 국제항해에 종사하지 않은 선박으로서 500톤 이상의 국내 화물선이며, 기간은 2002. 7. 1부터 2008. 7. 1까지 선박별 연차적으로 시행해왔다(IMO, 2002). AIS의 보급이 확대되면서, 육상에서도 선박 안전운항을 도모하기 위한 선박 모니터링 관련 서비스에 대한 수요가 높은 실정이다.

서비스 제공을 위해서는 기본적으로 통신망과 서비스 개발 기술이 뒷받침 되어야한다. Maritime VSAT(Very small aperture terminal)과 같은 광대역 위성 인터넷 서비스가 이미

전 세계 주요 항로에서 서비스되고 있으며, 앞으로도 육해상 간 통신망이 더욱 안정적으로 구축되면, 서비스 요구에 따른 대응이 충분히 가능해질 것으로 예상된다.

서비스 제공 환경과 관련하여, 최근 등장한 스마트폰이나 태블릿 PC와 같이 온라인에 상시 접속 가능한 모바일 장치와 서비스가 발전함에 따라, 인터넷을 사용할 수 있는 다양한 장치에서 멀티미디어 웹 응용서비스에 대한 수요가 높아지고 있다. 이를 지원하는 기술로써, 차세대 웹 표준인 HTML5는 별도의 플러그인 없이 멀티미디어 응용을 개발할 수 있는 API(Application Program Interface)를 제공하고 있다(이, 2010). 표준화 일정은 2007년 5월 HTML5와 Web Forums 2.0 스펙을 채택하는 것으로 시작하여, 2012년 정식 권고안이 나올 예정이다(W3C, 2011). 표준화 진행과 더불어 HTML5를 기반으로 다양한 응용이 이미 개발되어 사용되고 있다(Word, 2011).

본 논문에서는 위와 같은 최신 경향에 대응하여 웹기반 환경에서 HTML5를 사용하여 장치 독립적으로 언제 어디서든지 사용할 수 있는 선박안전항해를 위한 AIS선박(AIS를 장착

† 주저자, 교신저자 : sjlee@hhu.ac.kr 051)410-4578

\* 연회원 : wookis6766@naver.com 051)410-4888

한 선박)의 모니터링 시스템을 구현해보았다.

본 논문은 이러한 HTML5의 이점을 이용하여, 기존의 서로 다른 웹 기반 플랫폼에서 개발된 선박안전정보를 통합 서비스 할 수 있는 인프라를 구축하고, 또한, 비용이 들지 않는 상용 맵에 디스플레이 함으로써, 기존 전자해도 기반 응용보다 일반 사용자를 포함한 다양한 계층에게 제공할 서비스 개발의 가능성을 시험하는데 그 목적이 있다.

## 2. 차세대 웹표준과 AIS 기본개념

### 2.1 차세대 웹 표준 분석

W3C(World Wide Web Consortium)은 웹 표준을 개발하는 조직으로서 웹 언어 표준인 HTML을 개발하고 연관 기술을 연구한다. W3C에서는 1997년, HTML 4.0을 발표한 이후 HTML은 약간의 오류 수정을 제외하고는 그에 관한 개선 및 새로운 버전에 대한 작업을 진행하지 않았고, 문법 검사가 강화된 XHTML(Extensible Hypertext Markup Language)과 같은 웹에 보조적이거나 다른 방향의 기술을 주로 개발했다.

지금 까지 웹 표준으로서 널리 쓰이는 HTML4.01은 발표된 지 10년이 넘었다. 때문에 퍼블리셔들은 언어와 브라우저의 제한으로 인하여 브라우저 자체적으로 새로운 기술이나 향상된 기능들을 사용하지 못하고, 다양한 응용 서비스를 개발하기 위해서는 ActiveX나 Flash 같은 외부 플러그인이나 각 브라우저 업체별로 서로 다른 기술에 의존해야 한다.

최근 몇 년 동안 다양한 플랫폼을 이용한 모바일 기기들이 출시되고 그러한 환경에서의 웹의 수요가 증가 하고 있다. 하지만 기존의 웹에서 동적인 환경을 구현하기 위해서 사용해야 하는 외부 플러그인들을 구동하기 위해서는 많은 자원을 필요로 한다. 더구나, 하드웨어 및 소프트웨어 사양이 다양하면서도 유저자원이 적은 모바일 기기들에서 구동하기란 매우 어렵다. 이러한 문제가 대두되면서 W3C에서는 결국 차세대 웹 표준 개발을하기로 결정하였고, WHATWG(Web Hypertext Application Technology Working Group)에서 제안한 기술을 그대로 수용한 HTML5를 승인하고 표준화 작업을 시작했다(전, 2010).

별도의 플러그인을 설치하지 않고도 자체적으로 동영상, 오디오, 그래픽렌더링(Canvas), 오프라인 웹, 강력해진 웹 폼, 로컬 자원에의 접근 등 다양한 기능을 지원하고 있다(HTML5, 2011). 이 점은 플랫폼 독립적인 웹 응용 서비스의 개발이 가능해지고, 기존 로컬 응용 어플리케이션을 일부 대체 가능함을 의미한다.

### 2.2 AIS

AIS는 적절한 장비를 갖춘 육상 기지국, 타 선박 및 항공기에 선박의 신원, 종류, 위치, 침로, 항해 상태 및 다른 안전정보를 포함한 정보를 자동으로 제공하고, 유사설비를 갖춘 선박에서 수신한다. 또한 선박을 감시 및 추적하고 육상기지국

과 정보를 교환 할 수 있는 기능을 갖춘 장비이다. 현재 SOLAS 규정에 의해 국제 항해 시 총 톤수 300톤 이상, 연안 항해 시 총 톤수 500톤 이상의 모든 선박에서 탑재 의무화가 되어 있다(IMO, 2002).

AIS는 선박 장비 간 통신 표준 프로토콜인 NMEA 0183를 사용한다. NMEA 0183의 모든 데이터는 8비트 ASCII 문자로 표현되며 각 유효 데이터필드는 ","로 구분된다.

`!AIVDM,1,1,A,1P000Oh1T1svTP2r:43grwb0Eq4,0*01<CR><LF>`

Fig. 1 The Example of AIS message

Fig. 1은 캡슐화 되어있는 AIS 데이터가 NMEA 0183 프로토콜로 전송되는 예를 보여준다. 각 필드의 의미는 다음과 같다.

- 1) AIS VDM(VHF Data-Link Message)임을 알림
- 2) 전체 메시지의 수
- 3) 전체 메시지중 번호
- 4) 서로 다른 메시지로 구분할 수 있는 순서 번호
- 5) AIS 채널
- 6) 캡슐화된 AIS 데이터
- 7) 채움 비트

이 중에 6번째 필드로 이 필드의 데이터는 데이터 전송량을 줄이기 위해 6비트의 바이너리 코드로 캡슐화 되어 있다. 이 바이너리 코드의 크기가  $101000_2$  보다 작을 경우  $00110000_2$  을, 클 경우  $00111000_2$  을 더한다. 이렇게 나온 데이터는 ASCII 코드에 대응되는 문자로 표현된다. Fig. 2는 6비트 바이너리 코드가 아스키 코드로 변환 하는 알고리즘을 나타낸다(IEC, 2001).

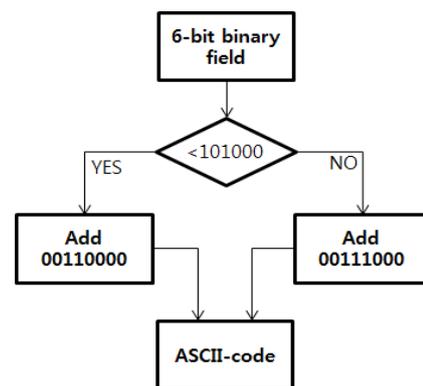


Fig. 2 Convert 6-bit binary field to ASCII-code of the symbol representing the 6-bit binary field

### 2.3 기존 웹기술의 한계와 웹소켓/웹워커 도입배경

일반적으로 기존 웹에서 새로운 정보를 화면에 표시하기 위해서는 웹 페이지 주소에 접근하여 웹 페이지 전체를 다운

로드하여 화면 전체를 갱신시키는 방법을 쓴다. 이것은 서버와 클라이언트가 동기적으로 동작하는 것으로 브라우저는 페이지 요청후 응답이 완료되기 전까지는 동작할 수 없다.

반면 Ajax(Asynchronous javascript and XML)를 사용할 경우 서버와 클라이언트 간 비동기적으로 동작하기 때문에 요청에 대한 응답이 완료되기 전에도 브라우저의 활동이 이루어질 수 있다. 또한 문서의 전체 요소를 각각의 객체로 보는 DOM(Document Object Model)을 활용해 사용자가 의식하지 않아도 페이지 내의 정보를 부분적으로 변경하는 것이 가능해진다. 하지만 기존의 웹과 마찬가지로 Ajax는 클라이언트의 요청에 대한 서버의 응답이 완료될 경우 연결을 지속하지 않고 바로 끊어버리는 연결 지속성에 대한 문제가 있다.

연결 지속성문제는 Comet이라는 Ajax에서 푸쉬(Push)를 구현하기 위한 웹 애플리케이션 모델로 해결이 가능하다. Comet은 기본적으로 Ajax 유사하게 동작하지만 요청에 대한 서버의 응답이 즉시 이루어지는 것이 아니라 서버 측 이벤트가 발생할 때 까지 대기하는 것이 가장 큰 차이점이다.

Fig. 3에서 Ajax와 Comet의 기본 동작을 보여준다. Comet의 동작 방식에는 크게 두 가지가 있는데 첫째는 롱폴링(Long Polling), 둘째는 스트리밍(Streaming)이다.

일반적인 폴링은 브라우저에서 주기적으로 서버에 요청하고, 서버는 즉시 그에 대한 응답을 하는 것이다. 롱폴링은 서버의 응답 후에 브라우저에서 즉시 재요청 하는 것으로, 서버가 응답하기 까지 시간이 일반적인 폴링에 비해 긴 편이다. 롱폴링은 연결이 끊기는 즉시, 연결을 시도하는 것으로 연결 지속성을 유지한다.

스트리밍은 서버의 응답이후 요청 없이도 연결이 지속되며 서버에서 능동적으로 응답이 반복적으로 지속되는 것이 가능하다. 둘 중 스트리밍이 가장 이상적이지만, 표준화 된 방식이 아니라서 브라우저 호환성에 대한 문제가 있다. 반면 롱폴링은 기존의 Ajax와 크게 다르지 않아 브라우저 호환성은 확보가 되어 있다. 하지만 요청-응답 과정에서 불필요한 오버헤드가 발생한다. 이점은 전송 데이터의 크기가 작고 반복적인 경우에 큰 단점으로 작용한다 (Wikipedia, 2011).

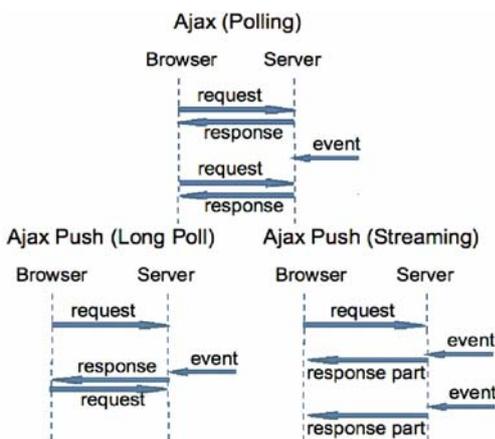


Fig. 3 Comparison of the transfer of Ajax and Comet

기본적으로 웹브라우저는 Single Thread로 동작한다. 때문에 적지 않은 데이터를 한꺼번에 수신하여 처리하거나, 복잡한 연산을 수행하는 경우에는 그 처리시간동안 브라우저가 마우스나 키보드의 입력에도 즉시 응답하지 않는 Freezing 현상이 발생할 가능성이 있다. 보다 복잡한 웹 어플리케이션을 구축하는데 있어서 문제점중 하나이다.

이러한 문제들을 해결할 수 있는 대안으로 제시되는 기술이 HTML5의 웹소켓(WebSocket)과 웹워커(WebWorker)이다. 웹소켓과 웹워커는 HTML5의 구현을 안정적으로 지원하는 중요한 연관기술로 다뤄지며 함께 개발되고 있다.

웹소켓은 브라우저에서 직접 소켓통신이 가능한 API를 제공하고, HTTP와는 다른 프로토콜을 사용하기 때문에 웹소켓을 구현하기 위해서는 별도의 API를 구현하고 서버를 구축하여야 한다. 웹소켓은 일반적인 TCP 소켓통신의 특징을 그대로 따르며 서버와 브라우저가 실시간으로 연결된 상태에서 독립적인 행동을 할 수 있다.

Fig. 4는 각각 폴링 방식과 웹 소켓을 활용하여 제작한 어플리케이션에서 발생하는 오버헤드의 크기를 보여주는 그래프이다. 각 A,B,C 그래프는 각각 1,000개, 10,000개, 100,000개의 클라이언트들이 접속하여 매 초마다 데이터를 수신할 때 발생하는 오버헤드의 크기를 나타낸다. Fig. 4에서, 클라이언트의 수와 데이터 전송 횟수가 증가할수록 기존의 AJAX 방식의 경우 오버헤드가 기하 급수적으로 늘어나며, 웹소켓을 사용하는 경우 이것이 크게 줄어드는 것을 확인할 수 있다 (Lubbers, 2009).

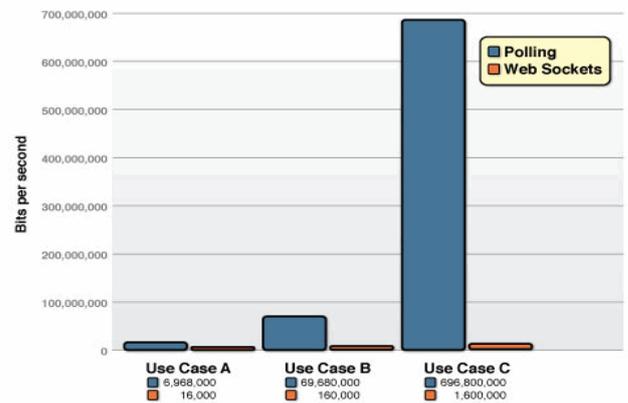


Fig. 4 Comparison of the unnecessary network throughput overhead between the polling and the Websocket application

웹워커는 멀티 스레드 프로그래밍을 웹 브라우저에서 구현한다. 웹워커는 데이터의 공유가 원천적으로 금지되어서 데드락 상태에 빠질 위험이 적다. 따라서 각 워커는 postMessage 메시지를 통해서 데이터를 전달하고 onmessage 이벤트로 수신한다. Fig. 5는 웹워커가 데이터를 송수신한 하는 개념도를 나타낸다.



Fig. 5 Data delivery concept of the WebWorker

또한 웹워커는 백그라운드에서 동작하기 때문에 복잡한 스크립팅이나 수학적 계산, 긴 지연시간이 발생할 수 있는 네트워크 액세스에 사용한다. 이를 효과적으로 사용할 경우 복잡한 연산중에도 자유로운 웹 브라우징이 가능해진다.

### 3. 시스템 설계 및 구현

본 논문에서는 설계하는 시스템은 크게 AIS 서버와 데이터베이스, 웹 서버와 웹 소켓 서버 그리고 웹 브라우저의 다섯 부분으로 나눌 수 있다. Fig. 6은 시스템의 전체 블록 다이어그램을 보여준다.

브라우저는 처음에 웹 서버로 접속해 웹 페이지와 스크립트를 받아온다. 그리고 현재 보고 있는 지역에 대한 정보를 웹 소켓 서버로 전송한다. 웹 소켓 서버는 접속한 브라우저를 등록하고 실시간으로 전송되는 지역 정보를 갱신한다. 정보를 제공하는 지역은 본 논문에서 다루고 있는 연구 장비의 지역적 한계와 연구 목적상 부산항 일대로 제한하였다.

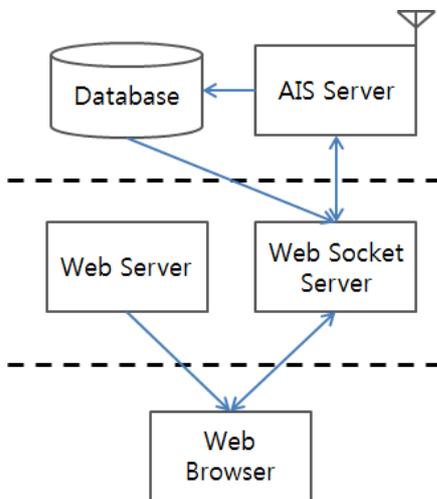


Fig. 6 System block diagram

AIS 서버는 AIS Receiver로부터 수신한 데이터를 AIS 표준 문서에서 제공하는 알고리즘에 따라 파싱(Parsing)하고 필요한 데이터를 수집하여 데이터베이스에 저장을 한다. 그리고 데이터베이스에 저장을 함과 동시에 웹 소켓 서버에도 메시지 알림(Message notification)을 보낸다. Fig. 7은 구현한 AIS서버 모니터와 데이터베이스 테이블의 일부를 보여준다. 데이터베이스는 MS SqlServer 2005, AIS 서버는 MFC로 구현하였다.

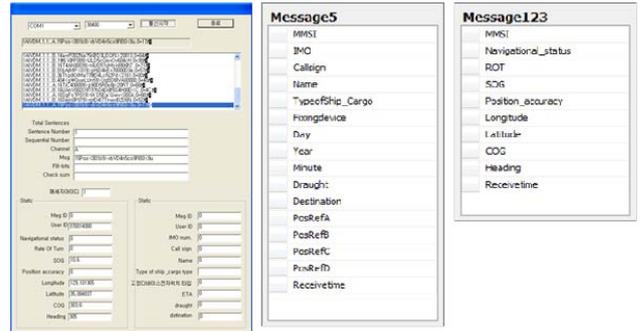


Fig. 7 AIS server monitor and database table

웹 소켓 서버는 AIS 서버로부터 메시지 알림을 받기 위해 실시간으로 TCP 소켓 연결을 유지한다. AIS 서버로부터 메시지 알림이 올 경우 서버는 메시지를 분류하고, 접속되어 있는 브라우저에 데이터를 전송하거나 기타 메시지를 전달하게 된다. 또한 브라우저의 요청에 따라 일정 시간 동안의 데이터를 요청받으면 데이터베이스로부터 받은 데이터를 한꺼번에 전달 할 수도 있다. 웹 소켓 서버를 구현하기 위해서는 WAS(Web application server)의 모듈 형태로 구현하는 것과, Stand-alone 서버로 구현하는 것이 있다. 본 논문에서는 자바로 제작된 jWebSocket을 활용해 Stand-alone 서버를 구현하였고, Apache 웹 서버를 사용하였다.

Fig. 8은 브라우저에서 실제 구현 화면이다. 사용한 브라우저는 Google Chrome 10이고 Google Maps API를 활용해 지도를 구현하였다. (정보보호를 위해, 상세 정보는 그림자 처리를 했다.)

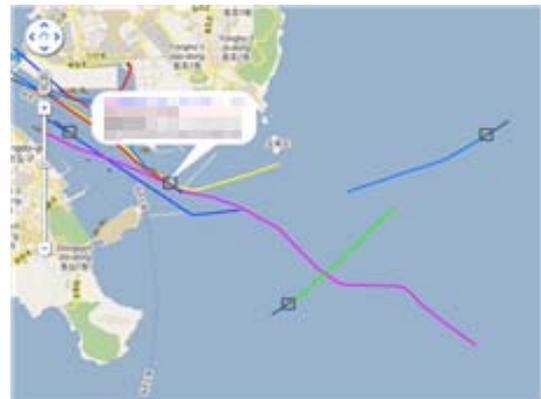


Fig. 8 Implemented screen using received information

### 4. 결 론

본 논문에서는 HTML5와 온라인 상용 맵을 기반으로 하는 AIS선박 모니터링 시스템을 구현했다. AIS수신정보 분석과 수신정보의 체계적 관리를 데이터베이스 구축을 하였고, 2012년 정식 권고안으로 표준화될 HTML5의 활용방안을 분석하고 구현해 보았다.

본 논문의 결과는 다양한 응용의 제작과 멀티 플랫폼 구현

이 가능한 HTML5를 활용하여 장치나 플랫폼, 즉 하드웨어적으로나 소프트웨어적으로 제약 없이 선박 안전 운항에 유용한 웹 기반 응용 소프트웨어를 제작할 수 있다는 것이다.

또한, 본 실험의 결과는 기존 국토해양부에서 운영하고 있는 AIS집약관리시스템과 연계하여 국가적으로 활용하거나, AIS를 이용한 국제적 연계시스템을 구축하기 위한 인프라로 활용할 수 있다.

향후 연구 과제로서 본 논문에서 구현한 시스템을 모바일 기기에서 정상적으로 동작하는 것을 구현하여 효과적인 멀티 플랫폼 서비스 제공 여부를 검증 하고, 다양한 기능과 형태의 응용을 구현해 볼 수 있다.

## 후 기

본 논문은 울산광역시 중소기업센터에서 주관하는 광역경제권 연계협력사업의 지원에 의한 연구임.

## 참 고 문 헌

- [1] 박영수, 이형기(2009), “해양사고 통계분석을 통한 VTS 개선방안에 관한 기초연구,” 한국항해항만학회, 제33권 (집), 제8호, PP.519~524.
- [2] 이원석(2010), “HTML5와 모바일웹,” TTA저널 통권 제 128호, pp. 50-54.
- [3] 전중홍(2010), “차세대 모바일웹 표준과 미래,” TTA저널 통권 제128호, pp. 61-67.
- [4] HTML5(2011), “HTML5, A vocabulary and associated APIs for HTML and XHTML,” <http://dev.w3.org/html5/spec/>
- [5] IMO(2002), SOLAS CHAPTER V, Regulation 19 Carriage requirements for shipborne navigational systems and equipment
- [6] IEC(2001), IEC 61993-2:2001(E), pp. 84-112.
- [7] Lubbers, P., Greco, F(2009), “A Quantum Leap in Scalability for the Web,” <http://websocket.org/quantum.html>
- [9] W3C(2011) HTML Working Group, <http://www.w3.org/html/wg/#sched>
- [10] Wikipedia(2011), Comet, [http://en.wikipedia.org/wiki/Comet\\_\(programming\)](http://en.wikipedia.org/wiki/Comet_(programming))
- [11] Word(2011), WordSquared, <http://wordquared.com/>

---

원고접수일 : 2011년 7월 11일

심사완료일 : 2011년 8월 28일

원고채택일 : 2011년 8월 28일