

Web 2.0

주요 기술 트렌드 및 관련 이슈

윤 석 찬

channy@daumcorp.com

(주)다음커뮤니케이션 R&D센터



Agenda

■ 1. 닷컴 버블 후 기술 기반 성공 사례

■ 2. 웹2.0 기반 기술의 아이디어 및 철학

- 사람 중심 시맨틱 웹
- 오픈 소스 및 오픈 스탠다드
- 지속적인 공개 베타

■ 3. 기획자들이 알아야 할 10가지 기술 요소

- 1. 웹 표준(XHTML/CSS)
- 2. 브라우저 지원(Firefox, Safari)
- 3. 유니코드 (UTF-8)
- 4. 논리 주소 체계 (Logical URL)
- 5. 콘텐츠 신디케이션(RSS/Atom, RDF)
- 6. 오픈 API (REST, SOAP, Web Services)
- 7. 집단 지성(Folksnomy, Tag)
- 8. 가벼운 서비스 프레임워크(Python, Ruby on Rails)
- 9. 풍부한 사용자 경험(Ajax, Flex)
- 10. 확장 기능 (Firefox Extensions, Widget)

■ 4. 향후 웹2.0 기술 트렌드

- Ajax 기반 비즈니스 응용
- 사용자 중심 아이덴티티(Identity)2.0
- 개발자 중심 생태계 시스템
- Mashup, 플랫폼 협업
- 모바일 웹(MobileWeb) 2.0
- 소프트웨어 서비스(Software Service)

■ Bookmarks

■ Q&A

본 자료의 그림 중 많은 부분은 Web Services Journal에서
(<http://www.wsj2.com>) 인용하였습니다.

After Bubble:

■ Google Platform

- Page-rank: 사용자들의 Link를 기반한 페이지 순위 변동.
- Ad-rank: 사용자의 클릭에 기반한 광고 순위 변동.
- Long-tail: 다수의 소형 사업자나 개인도 간단하게 광고 시장에 진입.
- Web Services: 야후! 등 주요 기업에 검색 ASP 서비스

■ Amazon Platform

- Review: 사용자들의 상품평을 주요 콘텐츠로 다룸
- Long-tail: 잠자고 있던 다량의 상품 데이터를 취급
- Web Services: 소규모 중개상이나 개인들이 이용하는 EC ASP 서비스

■ Ebay Platform

- Reputation: 거래자의 신용도를 상품 거래 시 반영
- Long-tail: 잠자고 있던 다량의 개인 물품 교환 데이터를 취급
- Web Services: 소규모 중개상이나 개인들이 이용하는 EC ASP 서비스

■ User/Advertiser/ASP 서비스들이 직접 참여하여 서비스 품질 및 비즈니스 모델에 만족하는 플랫폼(Platform)으로서 성장

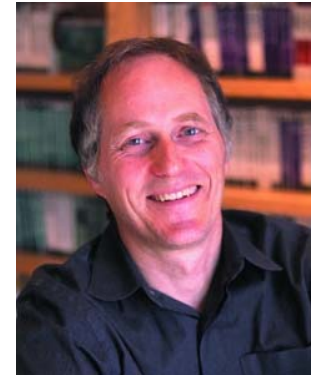
Web as Platform?

■ Definition

- I'm talking about the emergence of what I've started to call Web 2.0, the internet as platform.... it's about the way that web sites like eBay, Amazon, and Google are becoming platforms with rich add-on developer communities; it's about the way that network effects and data, rather than software APIs, are the new tools of customer lock-in;

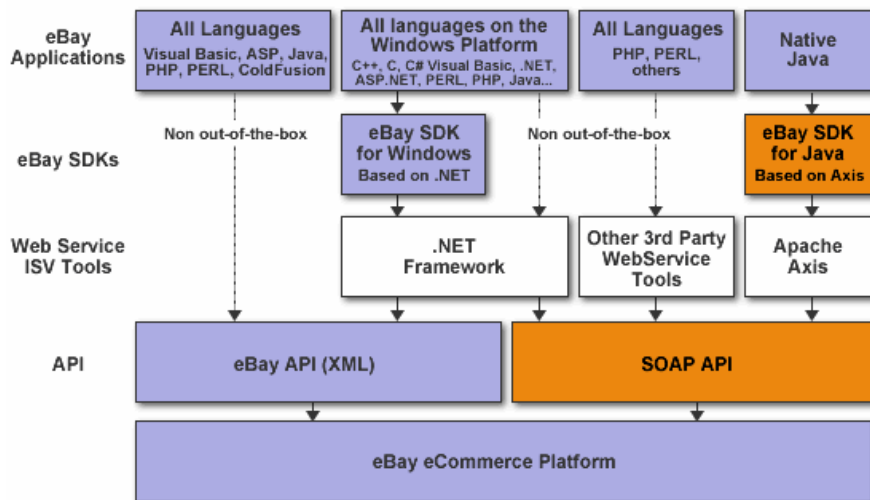
<http://www.oreillynet.com/pub/wlg/5630> Tim O'Reilly Sep. 21, 2004

- Web 2.0 design patterns"- architecting systems so that they get smarter the more people use them, monetizing the long tail via a combination of customer-self service and algorithmic management, lightweight business models made possible by cooperating internet services and data syndication, data as the "intel inside", and so on.



■ 플랫폼으로서 웹

- 웹이 소프트웨어 플랫폼처럼 스스로 자생하고 생존하는 시스템으로 진화 하고 있다.
- 데이터를 기반으로 Open API를 통해 개발자 커뮤니티를 생성하며 가벼운 비즈니스 모델로 모든 이용자들이 혜택을 받고 있다.



〈Ebay의 전자 상거래 웹 서비스 플랫폼〉

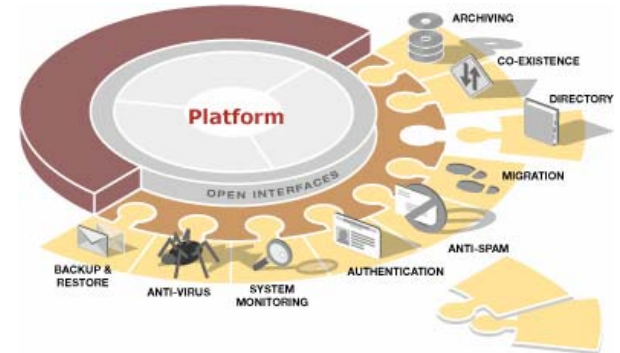


〈플랫폼을 이용한 비즈니스 방법을 다룬 책들〉

Web 2.0 기반 기술의 특징

■ 신 기술 = 신 개념

- 신기술 기반 서비스는 없다. 신개념 서비스만 있을 뿐!
- Web2.0의 특징은 기존의 PC 플랫폼 개념을 웹으로 옮겨 놓았다.



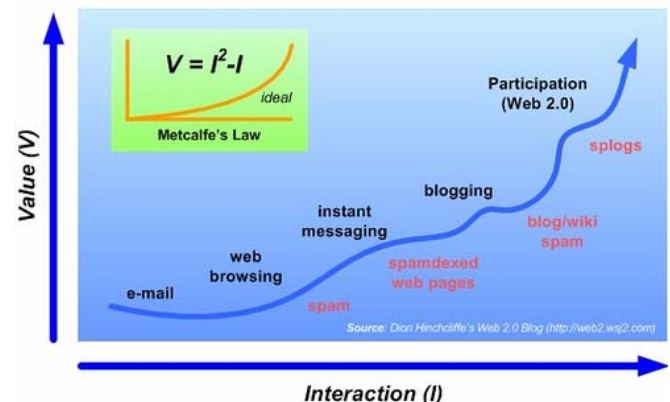
〈플랫폼 효과〉

■ 작은 것이 아름답다

- Web2.0 기반 서비스는 모두 핵심적이고 작은 서비스를 지향한다.

■ 오픈의 미학

- 정보 오픈은 네트워크 효과를 증대 시킨다.
- 사용자가 제어하는 지능이 있는 데이터를 기반할 수 있다. (정보 조직의 필요성이 더 크게 대두됨)



〈정보 오픈의 네트워크 효과〉

Concept 1: 인간 중심 시맨틱 웹

■ 기계 중심 Semantic Web

- Machine Readable Web (Structured Link)
- XML/RDF을 통한 데이터 구조화
- Web Services를 통한 데이터 전달 체계



■ 인간 중심 Semantic Web

- Human Readable Web (Logical Link)
- RSS/Trackback/Ping 등 Lightweight Syndication
- REST/XML-RPC 같은 가벼운 데이터 전달 체계

■ Web2.0에서의 Semantic Web

- Semantic Web은 궁극적으로 Machine Agent를 향하는 기술이었으나 결국 인간 중심 기술로 제공 되면서
- e.x) XHTML/CSS, RSS/Atom, REST, XML-RPC

Concept 2: 오픈 소스 및 오픈 스탠다드

■ 오픈 소스 (Open Source)

- 누구나 쉽게 개발에 참여할 수 있는 다양한 프레임워크 제공
 - Ex) LAMP (Linux+Apache+MySQL+PHP), Ruby on Rails, Python et. al
- 데이터를 다룰 수 있는 각종 오픈 소스 및 라이브러리의 무료 제공
 - Ex) Firefox Extensions, RSS Aggregator, JSON et. al
- 참여 문화를 이끄는 각종 도구
 - Ex) Wiki, Wordpress, Movable Type, Tattertools et.al

■ 오픈 스탠다드 (Open Standard)

- 서비스와 데이터를 다룰 수 있는 각종 오픈 API 제공
 - Ex) Google Maps, Flickr, Del.icio.us, Amazon, Ebay et. al
- 서비스와 데이터를 통용할 수 있는 각종 오픈 스탠다드 제공
 - Ex) RSS/Atom, Microformat, Attention.xml,

Concept 3: 지속적인 서비스 개발

■ 피드백과 진보를 담보하는 끊임 없는 베타 서비스

■ 가벼운 개발 방법론

- 프로토타입을 기반한 서비스 구현 및 테스트
- 성능 및 구조를 고려하지 않는 가벼운 개발 플랫폼 선택
- 기획/개발/디자인이 모두 참여하는 프로젝트 구조

■ 지속적인 베타의 덕목

- 오피니언 리더들과의 원활한 의사소통
- 참여자가 원하는 핵심 기능의 개선에 중점
- API 오픈을 통한 능동적인 참여 유도 및 혁신성 유지
- 참여자들의 디지털감성을 만족

10 Technology of Web 2.0

- 1. 웹 표준(XHTML/CSS)
- 2. 브라우저 지원(Firefox, Safari)
- 3. 유니코드 (UTF-8)
- 4. 논리 주소 체계 (Logical URL)
- 5. 콘텐츠 신디케이션(RSS/Atom, RDF)
- 6. 오픈 API (REST, SOAP, Web Services)
- 7. 집단 지성(Folksnomy, Tag)
- 8. 가벼운 서비스 프레임워크(Python, Ruby on Rails)
- 9. 풍부한 사용자 경험(Ajax, Flex)
- 10. 확장 기능 (Firefox Extensions, Widget)

1. Web Standards

■ 웹2.0과 웹 표준

- 웹2.0은 데이터를 다루는 기술이며 Web Design에서 Data Design으로 넘어 가야 한다. (웹 표준은 정보로서 웹을 다루는 가장 기본 방법)
- 웹 표준은 OS/브라우저/장치와 관계 없이 Forward Compatibility를 보장한다. (웹 표준은 Human/Machine Readable 한 기본 약속)

■ 웹 표준 개발 방법론

- 구조/표현/동작의 분리
 - 구조(Structure): XHTML을 통한 Semantic 정보 표현
 - 표현(Presentation): CSS를 통한 Design 및 Layout 표현
 - 동작(Behavior): DOM Scripting을 통한 UI 혁신
- 개발 동시성을 보장
 - 프로토 타입을 통해 기획을 공유하여 프로젝트 팀 기반 업무 추진 가능.
 - 영역 보장에 따라 기획/디자인/개발을 동시에 진행할 수 있다.
 - 빠른 변경 및 추가 기능 제공이 쉽게 진행 된다.

■ 웹 표준 왜 필요한가?

- 개발 속도 및 효율성 증대
 - 브라우저에 따른 고려가 없으므로 개발이 빠름.
 - 표준 구현에 능숙해 질 때 까지 약간의 비용 소요 되나 그 이후에는 개발 속도 향상
 - 개발자와 디자이너의 이중 작업 감소 및 영역 보장
 - 자동 Validation에 따른 QA 시간 감소.
- 웹 서버 및 네트워크 트래픽 감소
 - HTML 용량 감소 및 CSS/JS Caching을 통한 트래픽 감소
- 웹 페이지 재 사용성 강화
 - 하나의 템플릿으로 스타일 변경만으로 서비스 제공 가능
 - PDA, 텍스트, 장애인, 다국어 버전 스타일 적용
 - 구조/표현 분리에 따른 재 개발 비용 감소
- 글로벌 비즈니스를 위한 글로벌 스탠다드 준수
 - 1천만 명의 1%는 10만 명이나 1억의 10%는 1천만 명이다.

③ 접근성이 확보된 콘텐츠 구조

② 표현과 구조가 분리된 소스코드

2. Browser Support

■ 제2의 웹 브라우저 전쟁

- 현대 웹 브라우저 현황

- IE6 : 90% (웹 표준 준수 및 경쟁 브라우저 기능을 탑재한 IE7 출시)
- Firefox : 10% (오픈 소스 브라우저. 웹 표준 기반. 다양한 기능 및 확장 제공)
- Safari: 3% (맥 사용자에게 기본 제공. 웹 표준 기반.)
- Opera: 1% (웹 표준 기반. 다양한 브라우저 기능 제공)

- 브라우저간 웹 표준 지원 경쟁 (SVG, XML, Ajax et. al)

- 확장 기능(Extensions) 및 데스크탑 어플리케이션 대체 (Widget)

- 웹 브라우저 - 플랫폼으로서 웹을 만들어 내는 기본 영역

- Ex) Toolbar, Search Plugin, Extensions et. al

■ 웹 브라우저 지원과 웹2.0의 관계

- 웹 브라우저 지원은 고객의 기업 인지도 향상.

- 친환경 경영과 같은 것. 브라우저 호환을 통해서도 훌륭한 서비스 구현 가능

- 소수 사용자는 오피니언 리더다.

- 리눅스, 파이어폭스 및 맥킨토시 사용자는 인터넷 오피니언 리더.



3. Unicode

■ 문자 인코딩 문제

- 국내 대부분 웹사이트는 비 유니코드(EUC-KR)로 되어 있어 일본어/중국어 등 외국어 입력이 어렵다.
 - Ex)한글의 경우, 아햏햏 뵤 등의 글자 입력 지원 어려움
- 타 언어 OS 및 브라우저 사용자의 정보 접근성 제약
- Java, PHP, Python, MySQL 같은 개발 플랫폼에서 문자 인코딩은 항상 대하는 귀찮은 문제.
- XML 데이터 교환 등 Open API의 기본 문자 인코딩은 유니 코드.
- Ajax에서는 유니 코드(Unicode) 문자 인코딩만 지원.

■ 웹 2.0에서의 Globalization

- 인코딩 문제가 해결되면 L10N 처리 만으로 국제적인 서비스 제공 가능.
(Google, Technorati, Flickr)
- 세계적인 웹 2.0 서비스를 만들 수 있는 기본 바탕.

4. 논리적 주소 체계 (Logical URI)

■ URL이 Semantic 해야 하는 이유

- URL은 정보를 표현하는 기본 단위로서 Permanent 해야 한다.
- URL은 User readable, reliable, hackable 해야 한다.

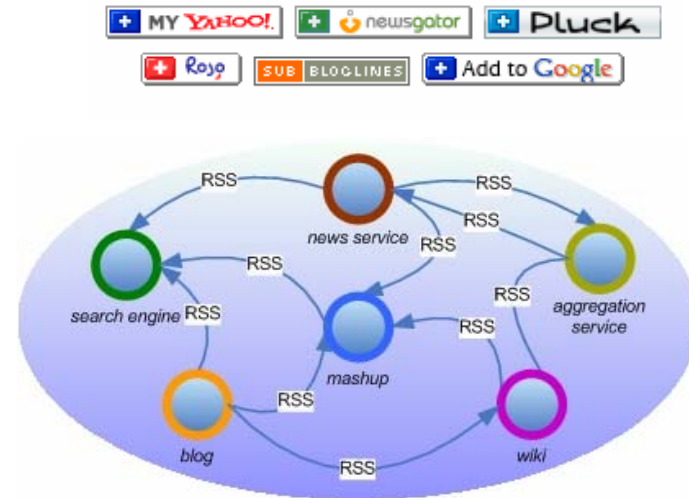
■ 웹2.0 기반 주소 체계

- URL은 구조적이고, 예측 가능, 해석 가능해야 한다.
 - Ex) <http://del.icio.us/settings/channy/profile>
 - Ex) <http://del.icio.us/channy/ajax/>
- URL에서 정보의 내용에 대해 노출 하라
 - Ex) http://www.digital-web.com/articles/web_2_for_designers/
- 프로그램 구조나 시스템을 노출 하지 말라.
 - Ex) <http://www.newsroom.ucla.edu/default.asp>
- 데이터베이스의 구조를 노출 하지 말라.
 - Ex) <http://www.newsroom.ucla.edu/page.asp?RelNum=6796>
 - c.f <http://del.icio.us/url/ce16162045b3eb98fdea3da77fc3d148> (hash값 사용)

5. Lightweight Syndication

■ RSS Everywhere

- Before: CDF(Channel Definition Format) :
포인트 캐스트의 몰락 (1997)
- Dave Winer's Userland가 Netscape의 RSS
포맷을 블로그에 처음 사용(2000)
- RSS(Really Simple Syndication)는 RDF의
킬러 어플리케이션으로 데이터를 교환 하는
방식을 전환 (2003)



Source: <http://web2.wsj2.com>

〈RSS는 정보 전달의 기본 단위 ©WSJ2〉

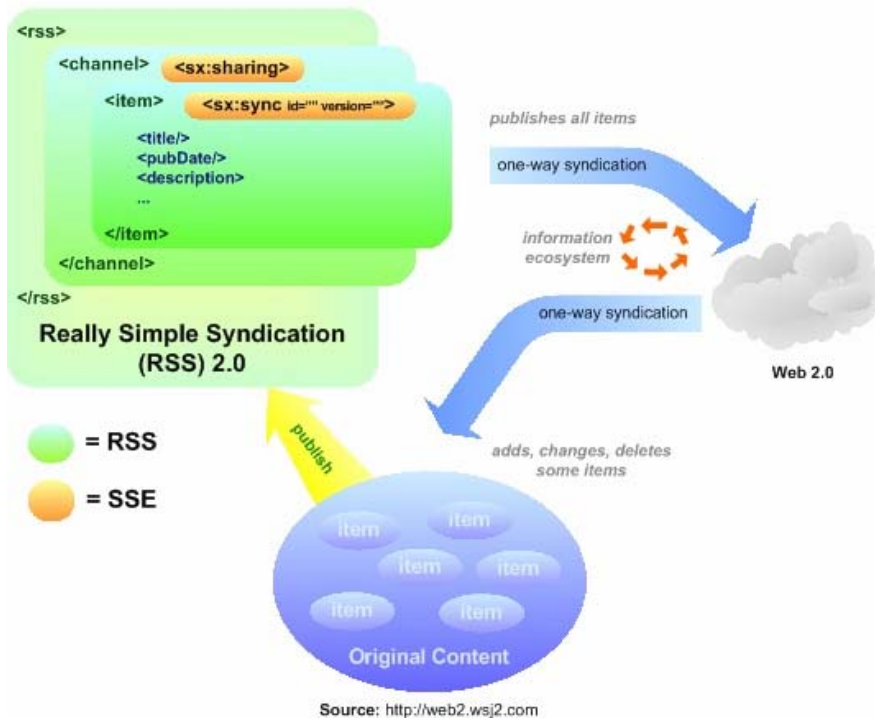
■ RSS를 통한 데이터 표현

- 병렬적인 데이터 제공: 모든 웹 페이지에 RSS
를 제공하라
- 멀티미디어, 광고, 검색 결과 등 모든 자료 전
달의 방식을 변환
 - Podcast, Ad Search
 - OpenSearch: A9에서 만든 RSS 기반 검색 결과 표
준 Ex) <http://opensearch.a9.com/>

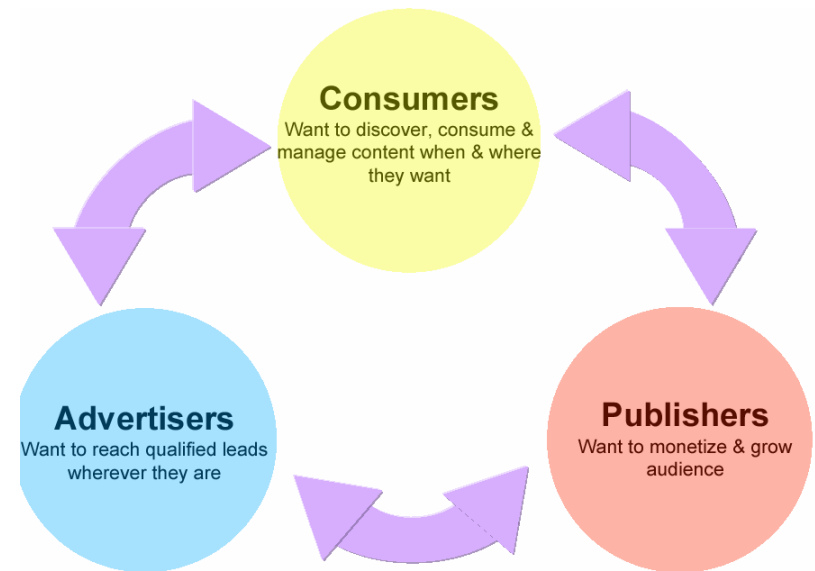
Total Usage of RSS



■ RSS: New Content Business Model



〈RSS를 통한 콘텐츠 확산 모델 ©WSJ2〉



〈RSS를 통한 콘텐츠 유통 모델 ©Yahoo!〉

6. Open API Model

■ Open API의 기본 철학

- 자신의 데이터 제어 (Editing, Viewing, Mixing, Matching)
- Open Data Format(호환 가능한 오픈 포맷 이용)

■ Open API의 기본 원칙

- 웹 페이지가 아닌 데이터를 디자인 하라
- 가려운 곳을 긁어라 (잘 안 되는 한가지 문제 풀기)
- 가능한 간단히 만들어라 (점진적 진화, 향상 시켜라)
- 사람이 먼저, 기계는 나중
 - 보기 좋고 읽기 좋게 하여 프로그래밍에 동참 시키기
- 광범위한 표준을 재사용 (표준을 준수하면 재창조 가능)
- 모듈화/임베딩 가능하게 만들어라.
- 개발, 콘텐츠, 서비스를 분산 시켜라

■ Open API 디렉토리

- WSFinder: <http://wsfinder.jot.com/WikiHome>
- Programmable Web:
<http://www.programmableweb.com/apis>



■ API 플랫폼 채택 방법

- 통신 표준: REST > XML-RPC > SOAP
- 언어: Javascript > PHP/Python/ASP > Java > C
- 기존 Open Format 재 활용
 - c.f iCal, RSS/Atom, Microformats.org

■ API를 만들어야 하는 열 가지 이유

- 1. 오피니언 리더 그룹에게 브랜드 전파할 수 있다
- 2. 사용자가 데이터를 소유할 수 있다.
- 3. 개발자 그룹과의 관계를 지속적으로 유지할 수 있다.
- 4. 커뮤니티를 통해 정보가 흘러 넘치게 한다.
- 5. API를 통한 문제 해결은 품질을 높여 준다.
- 6. 내부 데이터 재 사용 방법을 단순화 할 수 있다
- 7. 다른 사람이 서비스의 기능을 확장해 준다
- 8. 대체 자료 수집 방법 만들 수 있다.
- 9. 예상 못한 서비스가 나온다.
- 10. 서비스를 플랫폼으로 바꿀 수 있다.

7. Collective Intelligence

■ 참여를 통한 메타 데이터 수집

- Tagging as intelligence
 - 이미지, 동영상 등 데이터 검색 및 분류 용이.
 - 시간에 따라 콘텐츠 관심사 트래킹 가능
 - 다량의 데이터가 수집 시 태그와의 관계를 통해 분류 가능
- Auto Tagging?
 - 자동 태깅 기능을 부여하지 말라. (자동 태깅 기능은 사용자의 Attention을 무너 뜨린다.)
 - 태깅을 편리하게 하는 방법은 최대한 많이 제공하라. (태그 추천, 입력 UI 편리성)

■ Folksonomy

- 사용자의 태깅 데이터를 항상 계산하여 관련 태그를 추천하고 분류하라.(TagCloud)
- Top 10 같은 것을 제공하지 마라. 사용자에게 의해 만들어진 항상 최신의 인기 있는 모든 것을 제공하라. (편집 간여 금지)
- 사용자의 인터랙션을 최대한 많이 받을 수 있는 UI를 제공하라. (별점 기능)



8. Lightweight Framework

■ 개발 플랫폼의 진화

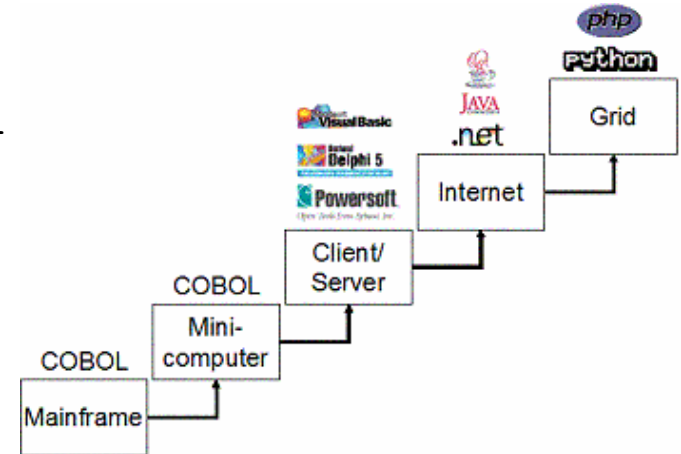
- Client/Server 환경에서 인터넷 기반으로 진화
- Java/.Net 보다는 빠르게 개발 가능한 Script 언어를 기반한 플랫폼 계속 증가
- 빠른 개발과 유지 보수에서 장점

■ 가벼운 개발 플랫폼

- MVC 모델 채용
- 성능 및 IDE 환경 제공
 - LAMP (Linux + Apache + MySQL + PHP/Python)
 - Ruby on Rails <http://www.rubyonrails.org/>

■ 가벼운 개발 플랫폼 장점

- 개발 환경 세팅 편리함
- 개발 라이브러리의 풍족함
- 개발자 pool이 넓음
- 빠른 프로토타입 가능함



〈개발 플랫폼의 진화 모델〉



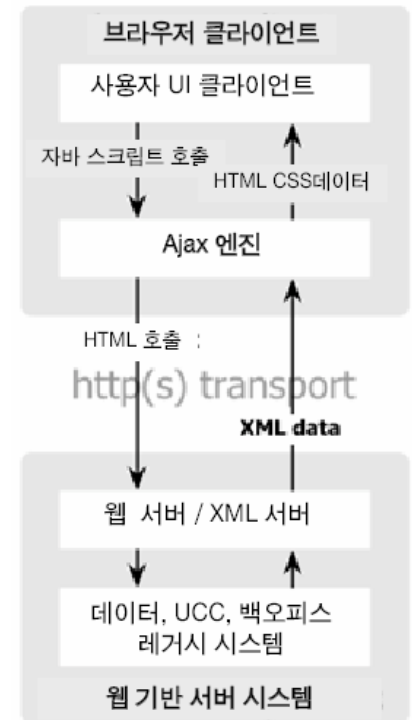
〈가벼운 개발 플랫폼 예〉

9. Rich Web Application

■ Ajax (Asynchronous JavaScript and XML)

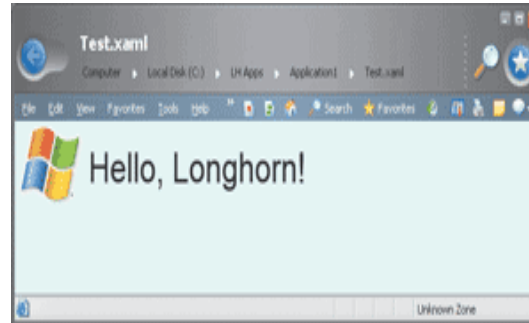
- Ajax의 특징
 - XMLHttpRequest와 자바스크립트를 이용한 비동기 데이터 교환
 - XML 및 XSLT를 통한 데이터 교환 및 이용
 - DOM을 함께 사용하여 다이나믹 표현 제공
 - XHTML 및 CSS를 이용한 표준 기반 표현 기법
- AJAX is not technology but approach (platform) like LAMP
- Ajax 기술은 블로그를 통해 기술 오피니언 리더들을 통해 전파되는 경향이 있다.
- Ajax 개발 상의 문제점
 - Browser 호환성
 - 브라우저 별로 상이한 DOM 핸들링 및 메모리 처리 방법
 - 어려운 자바스크립트 개발 및 디버깅 환경
 - 사용자 문제
 - 접근성 (장애인 및 비 Javascript 브라우저 등)
 - 사용성 (Back/Forward, Bookmark 등)
 - 기획의 문제
 - Ajax를 이해한 기획안이 나오기 힘들다
 - 사용자에게 친밀한 Application UI를 벤치 마킹 필요
- Ajax 사례 모음:

<http://channy.creation.net/blog/?p=156>



〈Ajax 처리 방식〉

■ Alternatives

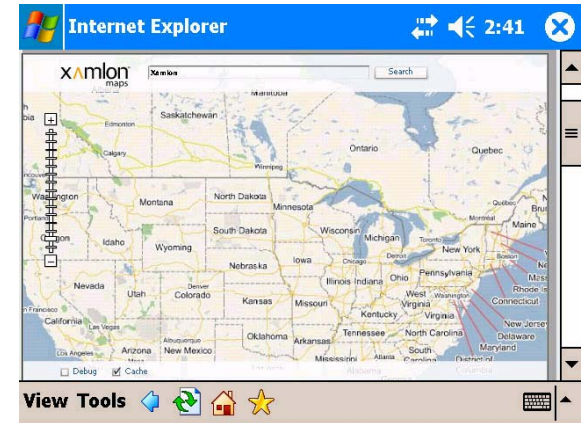


XAML (Windows Vista)

Macromedia Flex: <http://flex.macromedia.com>

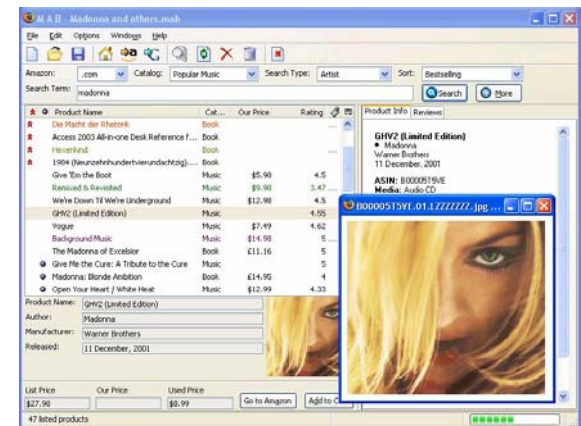


Yahoo! Widget: <http://widget.yahoo.com>



AFLAX (Asynchronous Flash® + XAML)

<http://www.xamlon.com>

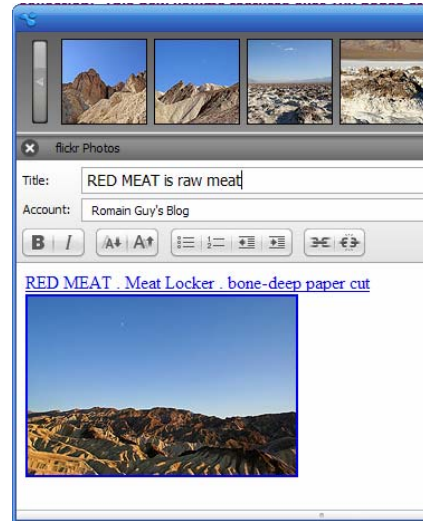


Mozilla Amazon Browser: <http://www.faser.net/mab>

10. 확장 기능

■ 웹 브라우저 확장 기능

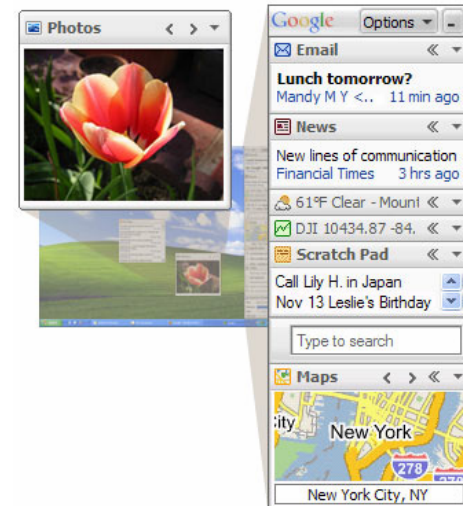
- Firefox Extensions
- Safari Widget
- Yahoo! Widget
- Windows Live Gadget
- Google Desktop Widget



〈Flock 브라우저를 통한 블로그 출판〉

■ Weblication: 대세 기술

- 웹을 더욱 다이나믹 하게 만든다.
풍부한 UI를 선보일 수 있다.
- 데스크탑과 경계가 모호해 진다.
- 고급 UI개발자 및 웹 개발자의 기
본 소양이 될 것이다.
- 다양한 RIA 기술 습득이 필수적.



〈Google Desktop Widget〉

Trend 1: Ajax 기반 비즈니스 응용

■ Ajax Platform Business

- Ruby on Rails (<http://www.rubyonrails.org>)
- Bindows (<http://www.bindows.net>)
- Backbase (<http://www.backbase.com>)
- Dojo (<http://dojotoolkit.org>)
- Sajax (<http://www.modernmethod.com/sajax/>)
- DWR (<http://getahead.ltd.uk/dwr/>)



Trend 2: 개발자 중심 생태계 시스템

■ 개발자 생태계 시스템?



- <http://developer.yahoo.net>
- 자사 API를 기반한 개발자 육성
- 자사 API를 기반한 Mash-up 서비스 육성

■ 대표적인 활동

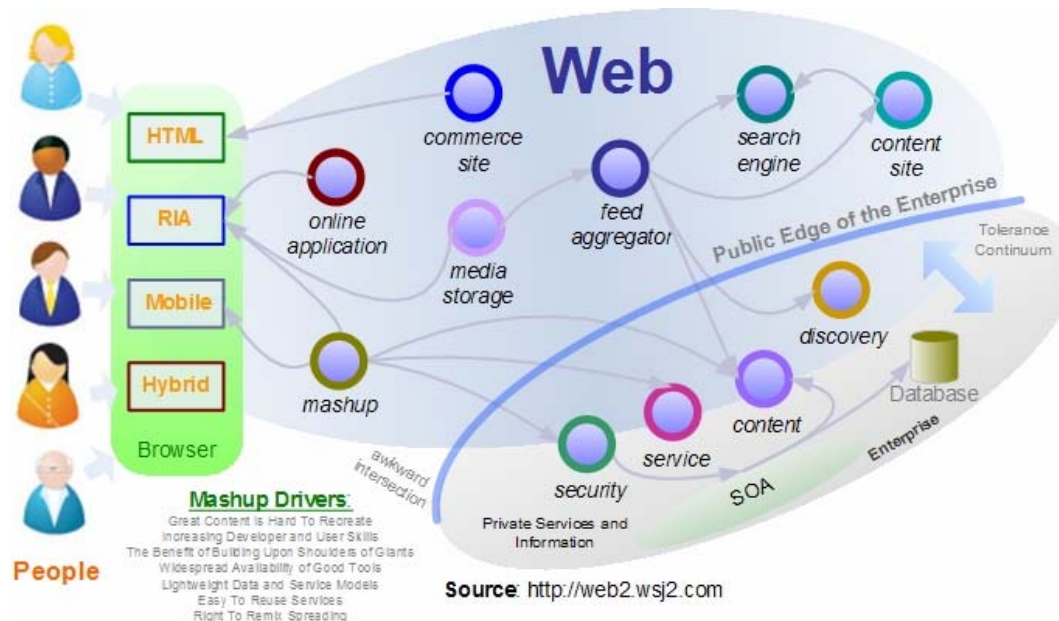
- API 어플리케이션 및 Mash-up 서비스 공개
- 오픈 소스 코드 제공 (<http://code.google.com>)
- 오픈 소스 공헌 (이용 프로젝트 개발자 채용)
- 개발자 블로그 운영을 통한 홍보
 - <http://sandbox.msn.com>
 - <http://labs.google.com>

Trend 3: Mashup, 플랫폼 협업

■ Mash-up 이란?

- Open API를 조합하여 전혀 다른 서비스를 개발하는 모델
- Mash-up 사례: <http://programmableweb.com/mashups>

■ Mash-up Eco-system



Trend 4: 사용자 중심 Identity2.0

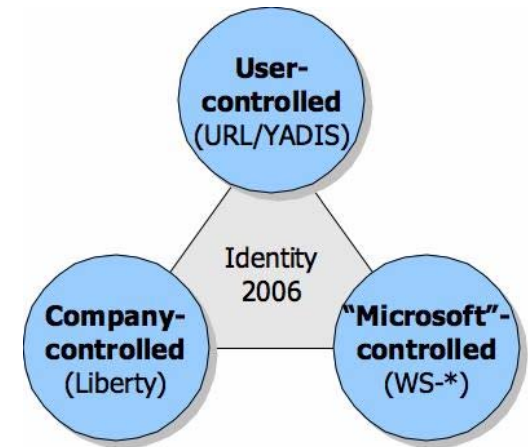
■ 웹 사이트 중심에서 사용자 중심으로 아이덴티티 관리 변화

- 원리 : URL을 사람에 대한 identifier로 사용하자

■ Think Changes !

- SXIP: Dick Hardt(CEO, founder), John Merrells
sxore, identity and reputation system
- LID : Johannes Ernst(CEO, founder), Joaquin Miller
mylid, identity and spam filtering
- OpenID : Brad Fitzpatrick
myOpenID, livejournal, OpenID comment for MT,WP
- YADI: OpenID, LID
- I-names :Drummond Reed
2idi(I-name), *universal private address* for personal

■ User-Open URL based Identity with Strong Authentification



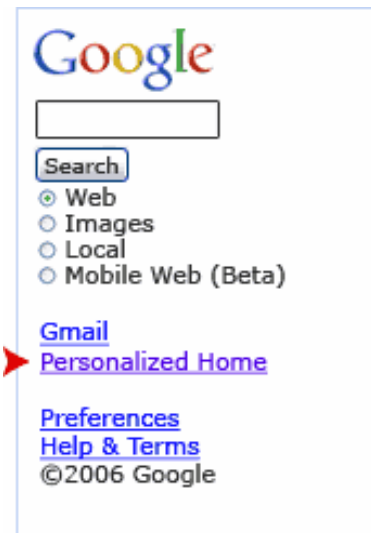
Trend 5: Mobile Web 2.0

■ WML is gone!

- 표준 기반 모바일 웹 브라우저
 - 해외: Opera, Minimo
 - 국내: 인프라웨어, 지오텔
- VoIP : Skype
- SMS: <http://textpayme.com>

■ 모바일 2.0 서비스

- Google Mobile
- Yubnub



■ Tracking Mobile 2.0

- MobileCrunch <http://mobilecrunch.com/>

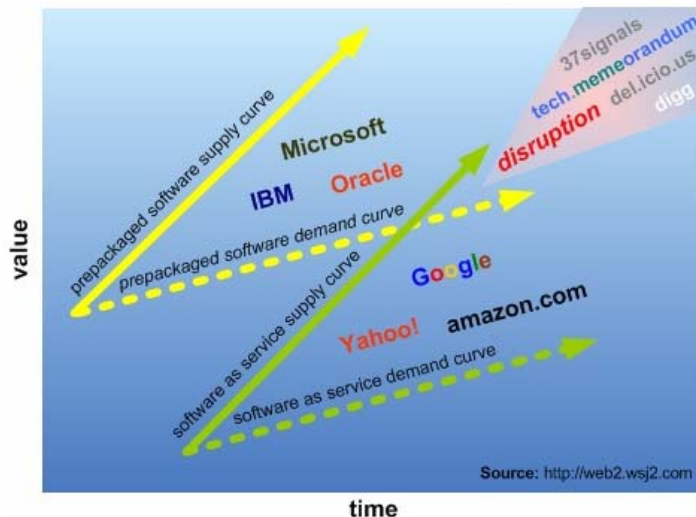
〈모바일웹2.0 사례〉

Trend 6: 소프트웨어 서비스

■ 웹과 데스크탑의 파괴(Disruption)

- Ajax 기반 Office
- Java 기반 Thinkfree
- Windows Live

■ 소프트웨어 수요/공급을 통해 본 웹 2.0 서비스 모델



The New Attention Focus: Web 2.0 Desktops

The go anywhere
information and
service pull model
browser

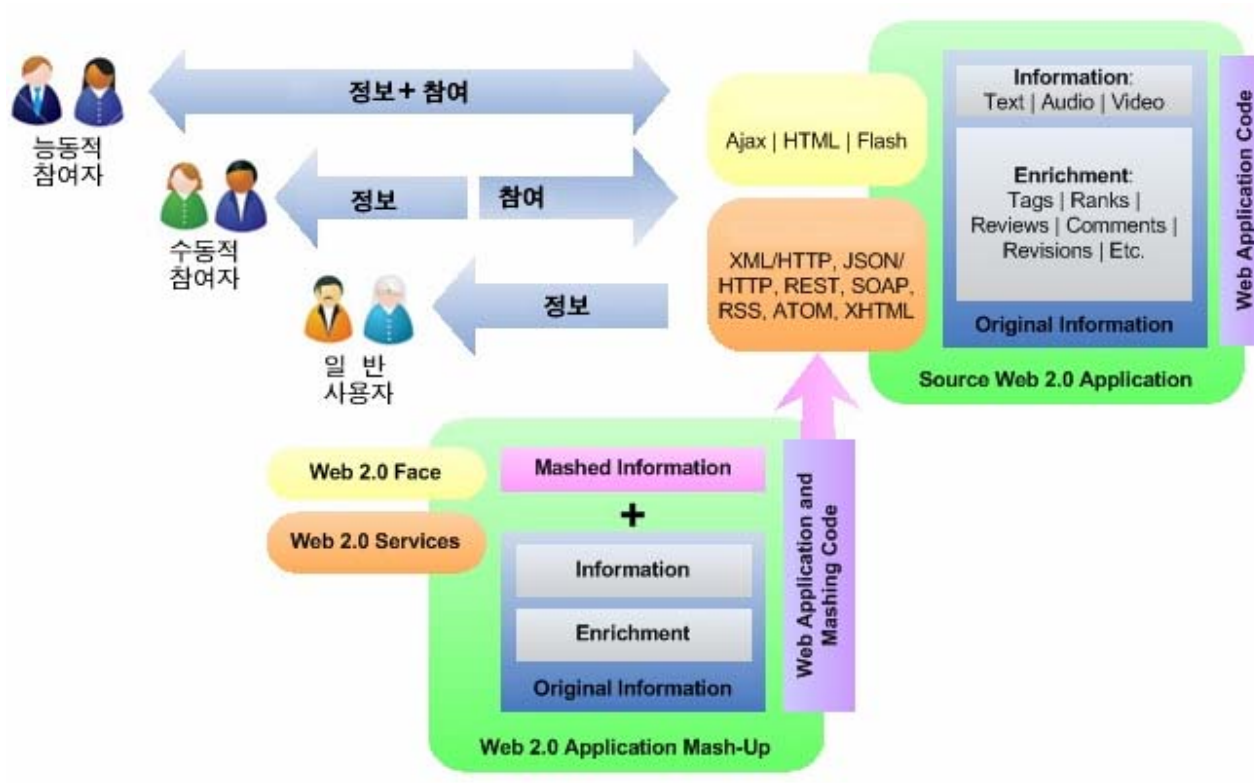


Examples:
Netvibes
Live.com
Pageflakes



〈웹2.0 기반 데스크탑 점이 현상〉

웹 2.0 기술 모델



Bookmarks

-Linkto: <http://channy.creation.net/web2con>

-mailto: channy@creation.net