

Web 2.0 Tutorial

2006.06

(주)인프라밸리 서비스개발1팀

문 서 이 력

Date	Updated Contents	Updated By	Version
2006-06-26	초안작성	오지훈	0.1
2006-07-03	사례사이트정리, 용어정리 추가	오지훈	0.2

Table of contents

1. Web 2.0 개요	4
1.1 Web의 발전사에 따른 Web 2.0의 위치.....	4
1.2 Web 2.0 개념정의	5
1.3 Web 1.0과 Web 2.0의 비교	6
1.4 Web 2.0 용어정리	7
2. Web 2.0 관련기술	8
2.1 롱테일 (Long Tail)	8
2.2 RIA (Rich Internet Application).....	9
2.3 RSS (Really Simple Syndication).....	11
2.4 OPML (Outline Processor Markup Language)	12
2.5 Trackback / Trackback Ping	14
2.6 Mashup Service	15
2.7 AJAX (Asynchronous Javascript And XML).....	16
2.8 Adobe FLEX	17
3. Web 2.0 특징	20
3.1 웹을 플랫폼으로 생각한다.	20
3.2 집단 지성을 활용한다.....	21
3.3 차별화된 데이터로 승부한다.....	22
3.4 소프트웨어 릴리스 주기의 혁신을 꾀한다.	22
3.5 가벼운 프로그래밍 모델을 사용한다.....	23
3.6 단일 디바이스 수준을 넘어선 소프트웨어를 지향한다.....	24
3.7 풍부한 사용자 경험을 제공한다.....	25
4. Web 2.0 사례 사이트 분석	26
4.1 del.icio.us (del.icio.us), 딜리셔스.....	26
4.2 Flickr (www.flickr.com), 플리커	28
4.3 Wikipedia (www.wikipedia.org), 위키피디아	30
4.4 싸이월드 태깅서비스 (www.cyworld.com)	32

4.5 Writely (www.writely.com), 라이틀리	34
4.6 Google 개인화 사이트 (www.google.com/ig)	36
4.7 피코디 개인화 사이트 (http://www.pcodi.com/)	38
5. 맺음말	39
5.1 사회적 네트워크의 형성	39
5.2 신규 사업기회 창출	39
5.3 사용자 언론 영향력 확대	39
Appendix. Web 2.0 용어 정리	40
- AJAX (Asynchronous Javascript And XML), 아작스, 에이잭스	40
- Collective Intelligence, 집단지성, 집단지능	40
- Content Syndication, 콘텐츠 신디케이션, 콘텐츠 수집	40
- FOAF (Friend Of A Friend)	40
- Folksonomy, 폭소노미	41
- Long Tail, 롱테일	41
- Mashup, 매쉬업, 혼합	41
- Niche Market, 니치마켓, 틈새시장	41
- OPML (Outline Processor Markup Language), 개요처리언어	42
- Personalization, 개인화	42
- RIA (Rich Internet Application)	42
- RSS (Really Simple Syndication / Rich Site Summary)	42
- Tag, 태그, 꼬리표	43
- Trackback, 트랙백, 원격댓글	43
- UCC (User Created Content), 사용자 제작 콘텐츠	43
- Wiki, 위키	43
- XFN (XHTML Friends Network)	44

Web 2.0 Tutorial

작성: 오지훈 (sonny@infravalley.com), 몽키몽키(cache798@naver.com)

(주)인프라밸리 기술연구소 서비스개발1팀

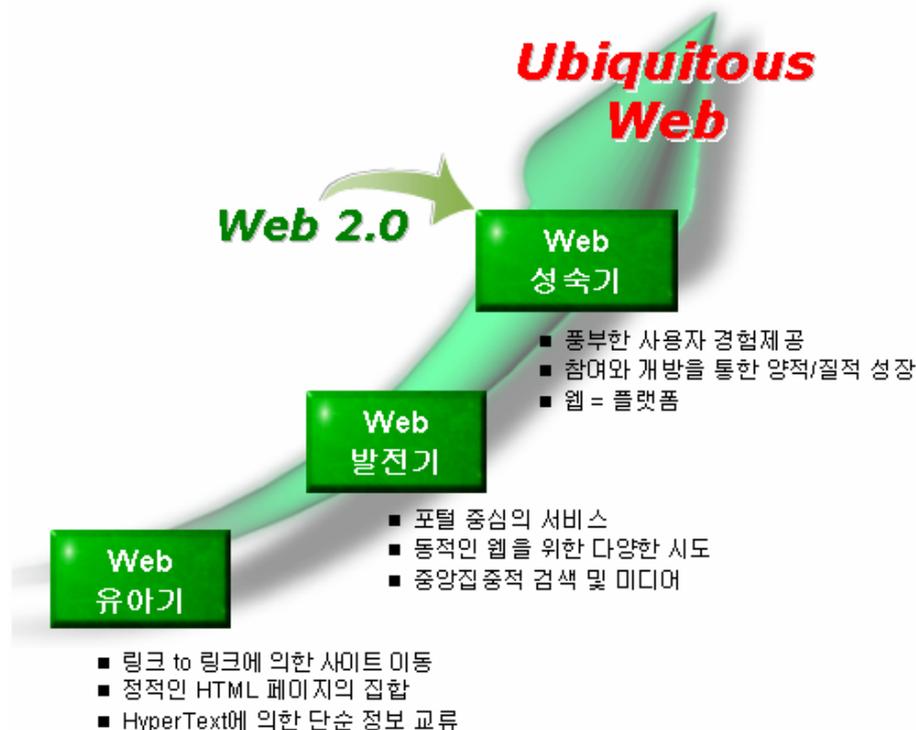
본 문서는 프로젝트 수행도중 틈틈히 시간을 내서 작성된 문서입니다. 개인적인 노력과 시간투자에 의한 산출물인 만큼 배포하실때는 꼭 작성자와 출처를 명시해 주시기 바랍니다.

1. Web 2.0 개요

1.1 Web의 발전사에 따른 Web 2.0의 위치

지금까지의 웹을 발전사를 정리해 보자면 크게 3 부분으로 구분된다.

정적인 HTML 페이지의 집합으로 링크와 링크간의 하이퍼링크 체계를 통한 사이트 이동이 주류를 이루어 단순정보 교류가 가능한 정도의 웹, 이를 “Web 유아기”, 포털 중심의 중앙집중적 서비스 방식으로 동적인 웹을 위한 다양한 시도가 있었던 시대의 웹, 이를 “Web 발전기”, 개방된 환경에서 자유로운 자발적인 사용자 참여를 통해 이루어진 웹 상의 플랫폼을 지향하는 Web 2.0 환경이 주축이 된 웹, 이를 “Web 성숙기”로 각각 칭해보자.



(그림) 웹의 발전사

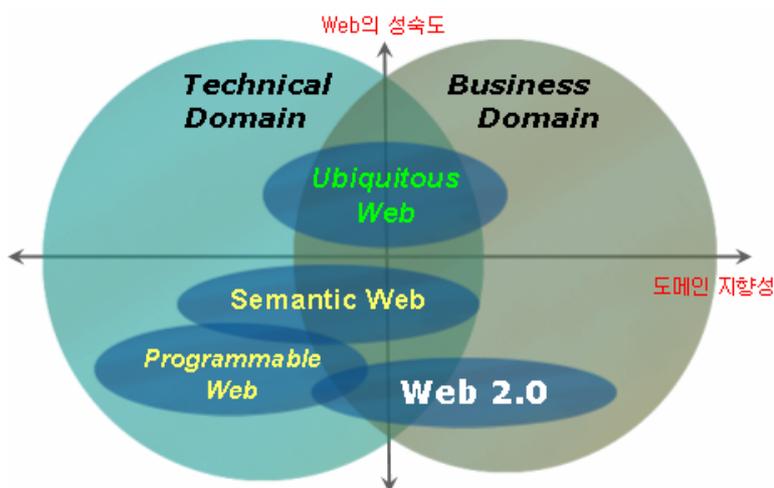
Web 2.0 성숙기에 위치한 Web 2.0의 특징을 살펴보면 다음과 같다.

- 기존의 웹은 기존의 정적이고, 포털 사업자에 의해 좌지우지되는 중앙 집중화된 공간인 반면에, Web 2.0은 개방된 환경하에서의 사용자 참여를 통해 이루어진 분산화되고 자율적인 공간이다. (Web 2.0 = 개방된 환경 + 사용자 참여)
- 웹이 궁극적으로 자동화와 지능화가 안정적으로 다져진 기반 위에서 사용자에게 보다 윤택하고 편리한 환경을 제공하는 시맨틱 웹 또는 유비쿼터스 웹을 지향한다. Web 2.0은 이 목표를 위한 중간 산물중의 하나이다.
- 머지않아 Windows와 같은 OS가 사라지지 않을까? 웹을 지원하는 냉장고, TV, 이외의 가전들을 통해서 Windows가 수행했던 여러 일들이 별도의 Desktop없이 이들을 통해서 수행될 수 있을 것이다. (웹의 OS화, 전통적인 플랫폼과 웹 플랫폼의 대결)

1.2 Web 2.0 개념정의

Web 2.0을 정확히 이해하기 위해서는 그 정의에 연연하기 보다는 Web 2.0이 어떠한 연유로 대두되었지 인지한 후, 그 특징을 하나씩 공감해 나가는 것이 첩경이지 않을까?

Web 2.0 소속 도메인을 Web의 성숙도와 도메인 지향성 측면에서 다음과 같이 표현하여 보았다. 궁극적으로 Ubiquitous Web으로 가기 위한 비즈니스 도메인에 편중된 중간과정의 산물이다. Web 2.0 이후로 아님 지금도 진행중인 프로그래밍의 대상으로써의 웹 (Programmable Web), 온톨로지, 웹의 지능화 관점에서 바라본 기술적 도메인의 Semantic Web 등의 개념이 어느 정도 자리를 잡을 것으로 생각한다.



(그림) Web 2.0 도메인

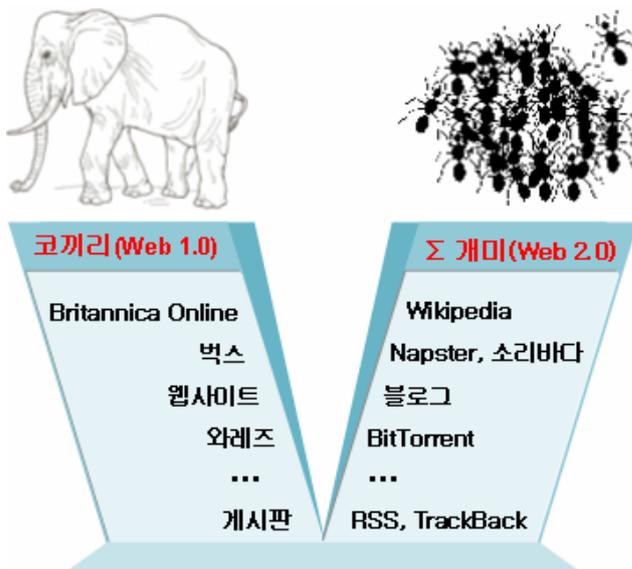
이를 종합해 볼 때 Web 2.0은 다음과 같이 정의할 수 있다.

- Web 2.0은 닷컴 붐과 이후의 살아 남은 업체들을 분류하고, 새로 등장하고 성장한 업체들의 공통점들을 총칭하는 용어로, 차세대 인터넷을 대표하는 용어로, 새로운 비즈니스 동향으로, 새로운 웹의 트렌드를 지칭하는 용어로 사용된다.
- Web 2.0은 바로 “플랫폼으로서의 웹”을 지향한다. 기술적으로는 기존에 진행되던 다양한 XML 응용, 웹서비스 응용, 시맨틱 웹 응용 등과 같은 차세대 웹 응용 기술들을 포괄하면서, 여기에 “개방”, “공유”, “협업”, “참여”라는 4가지의 네트워크 기반의 문화적인 키워드들을 결합시킨 개념이다.
- Web 2.0은 웹과 인터넷 비즈니스를 사용자 중심의 관점에서 재분석하여 새로운 기회를 찾기 위한 재발견된 기술적, 비즈니스적 트렌드이다.

“Web 2.0을 정확히 이해하기 위해서는 그 정의에 연연하기 보다는 Web 2.0이 어떠한 연유로 대두되었지 인지한 후, 그 특징을 하나씩 공감해 나가자.”

1.3 Web 1.0과 Web 2.0의 비교

“Web 1.0 또는 Web 2.0 응용의 구분 기준은 사용자 참여가 바탕이 된 개방된 자율적 환경에서 운용이 되고 있는지의 여부이다.”



(그림) Web 2.0 서비스와 Web 1.0 서비스 비교

Web 1.0과 Web 2.0을 비교해 보자면 다음과 같다.

- Web 1.0 서비스는 사이트 운영자에 의해 중앙 집중화된 환경하에서 정적으로 운영이 되지만, Web 2.0 서비스는 사용자 참여에 의해 자율적 분산환경에서 운영되므로

점차 양적/질적으로 데이터가 풍부해 진다.

- Web 1.0 서비스의 경우 포탈상에서 사용자 취향 및 의도에 맞게 조정이 불가능 하지만, Web 2.0 서비스의 경우에는 플랫폼상에서 운영되고 있는 서비스들은 사용자가 원하는 의도에 맞게 조정이 가능하다.
- Web 1.0은 기술중심으로 대부분이 OS/브라우저에 종속성을 가지고 있지만, Web 2.0은 사람이 중심이 되는 참여와 공유의 컨셉을 바탕으로 OS/브라우저에 상관없이 기능 구현이 가능할 뿐만 아니라 필요에 따라서는 사용자들에 의해 확장 가능하다.
- 브리태니커, 백스, 와레즈 사이트가 Web 1.0 서비스이고, 위키피디어, 냅스터, 비트 토런트가 Web 2.0 서비스인 이유는 Web 1.0 서비스의 경우 Web 2.0 서비스의 기본 컨셉인 개방, 협력, 참여, 공유와 같은 요소들의 지원이 미비하기 때문이다.

브리태니커 백과사전은 사이트 운영자에 의한 중앙집중적 관리로 운영되지만, 위키피디어는 사용자 참여에 의한 자율적 분산환경에서 운영되므로 점차 양적/질적으로 브리태니커를 능가하는 사전이 될 것이다.

1.4 Web 2.0 용어정리

다음의 Web 2.0 관련 용어들은 다음과 같은 4가지 기준에 의해 분류하였다. 복합적인 성격을 가진 용어들은 좀더 가중치가 큰 영역으로 배치하였다.

- 표준을 준수한 개방된 환경
- 손쉬운 데이터 서비스 및 공유를 위한 풍부한 사용자 경험 제공
- 협업에 따른 서비스 자동화
- 사용자의 활발한 참여에 따른 새로운 비즈니스

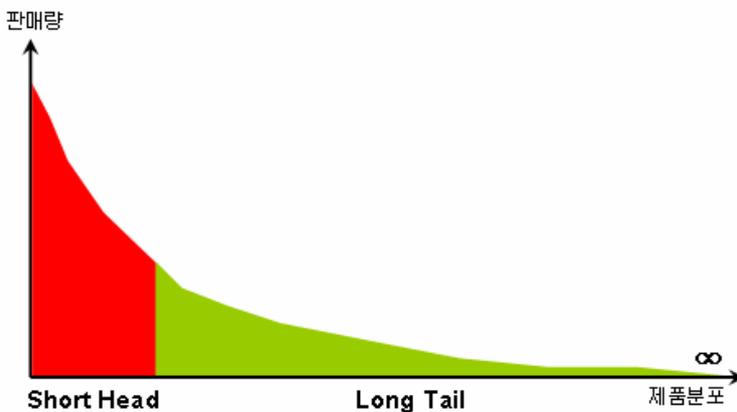


(그림) Web 2.0 용어분류

2. Web 2.0 관련기술

2.1 롱테일 (Long Tail)

롱테일이란 일반적으로 거의 판매되지 않는 니치(niche) 상품 판매액의 합계가 베스트셀러 상품 판매액의 합계를 상회하게 되는 현상을 말한다. 이것은 미국의 유력 월간지 와이어드(Wired)의 편집장 크리스 앤더슨이 주창한 것으로, 판매랭킹 순으로 판매액 곡선을 그리면 베스트셀러가 공룡의 머리(헤드)고, 니치상품이 긴 꼬리가 된다는 데서 이름 붙여졌다.

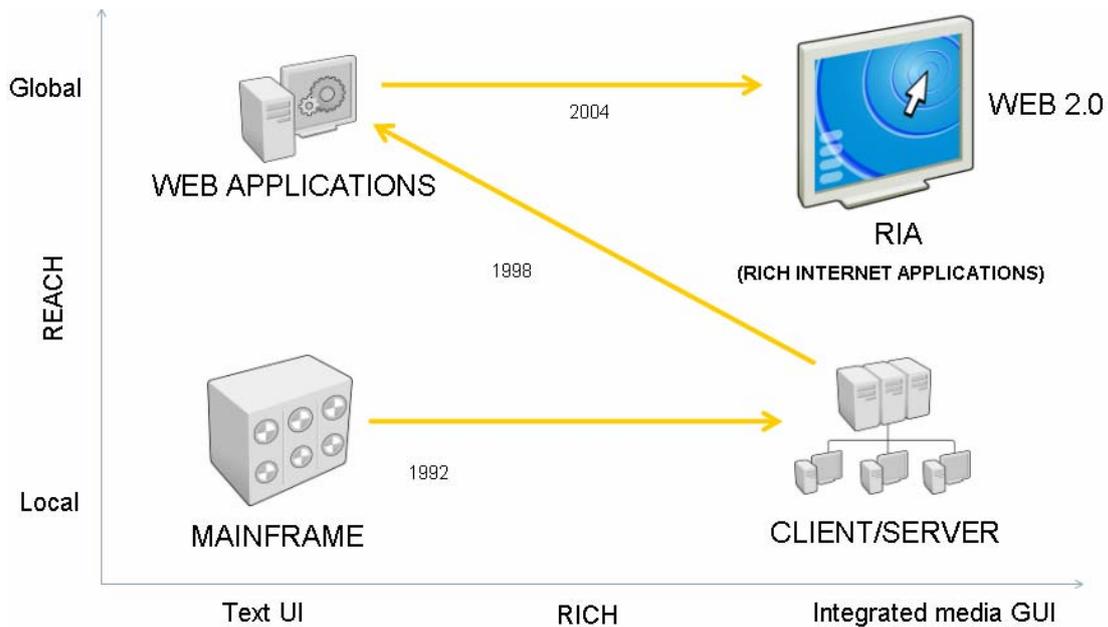


(그림) Long Tail 현상

- 롱테일(Long Tail)이란 이처럼 기존의 패러다임에서 사소한 것으로 간주되던 나머지 80%가 영향력이 크다는 것을 의미하는 용어로, 시장의 중심이 머리에 해당하는 소수의 히트상품에서 꼬리에 해당하는 다수의 니치(niche) 상품으로 움직여 가는 현상
- 롱테일을 이야기할 때 **머리**는 매출을 일으킬 수 있는 매장에 진열된 상품을 지칭하며, **꼬리**는 그 이외의 상품이나 개수를 의미한다. 예를 들어, 도서의 경우 베스트셀러 내에 드는 제품들이 머리에 속하고 그 외에 진열 자체가 불가능한 책을 꼬리이다.
- 롱테일이란 일반적으로 거의 판매되지 않는 니치상품 판매액의 합계가 베스트셀러 상품 판매액의 합계를 상회하게 되는 현상을 말한다. 이는 판매랭킹 순으로 판매액 곡선을 그리면 베스트셀러가 공룡의 머리(헤드)고, 니치상품이 긴 꼬리가 된다는 데서 이름 붙여졌다. 1km의 롱테일을 적분하면 공룡의 머리를 능가한다.
- 예를 들어 오프라인 서점은 진열 공간의 제약이 있기 때문에 진열 제품에 한계가 있는 반면에, 진열할 수 있는 상품의 개수가 무한대이며 다양한 사용자들이 접할 수 있는 온라인 서점 아마존은 230만 품목을 판매하고 있다. 이 때 판매 수 상위 13만 품목 이하의 서점 판매액을 합하면 전체 판매액의 57%에 달한다(2005).
- 롱테일 현상을 자사의 판매정책에 적용한 온라인 업체인 아마존(Amazon), 음악 판매 서비스인 애플 아이튠즈(itunes), 개인 간 베흥 시장인 이베이(Ebay). 이들의 공통적인 특징은 무엇일까?
→ **상품의 다양성 지향, 다양한 사용자 욕구 만족, 시공간의 자율성 등**

2.2 RIA (Rich Internet Application)

RIA는 미국의 매크로미디어사가 2001년 하반기 선보인 "플래시MX" 저작툴을 통해 선보인 개념으로, 웹 브라우저를 기반으로 하지만 마치 윈도우 프로그램 같은 풍부한(Rich) 기능을 보이는 것을 가능하게 하는 웹 클라이언트 기술이다.



(그림) Web 2.0과 RIA

- Web 2.0에서 중요한 기술적 이슈 중 하나가 Rich Client Application 이다. 기존의 Web Application의 편리한 배포, 관리, 저렴한 유지 비용 등의 장점과 Desktop Application 수준의 사용자 편의성, 효율적인 데이터 통신, 견고한 아키텍처 구현 등의 장점을 합친 것이 RIA 이다.
- RIA는 전체 페이지가 아닌 필요한 부분만 로딩해 페이지 부하가 적고 빠르며, 복잡한 디자인에 대한 렌더링 속도 저하 문제를 극복한다.
- Web 2.0을 구현하고 사용자 인터페이스를 위한 RIA 기술로는 매크로미디어사의 Flash와 마이크로소프트사의 ActiveX, 자바애플릿 기술을 들 수 있다. Ajax, Flex, Widget, Dash Board를 비롯한 Firefox의 확장도 RIA 기술로 분류할 수 있다.

RIA라는 용어는 Web 2.0 시대에 새로이 생성된 것이라고 생각하지는 않는다. RIA는 상대적인 개념이 존재한다고 생각한다.

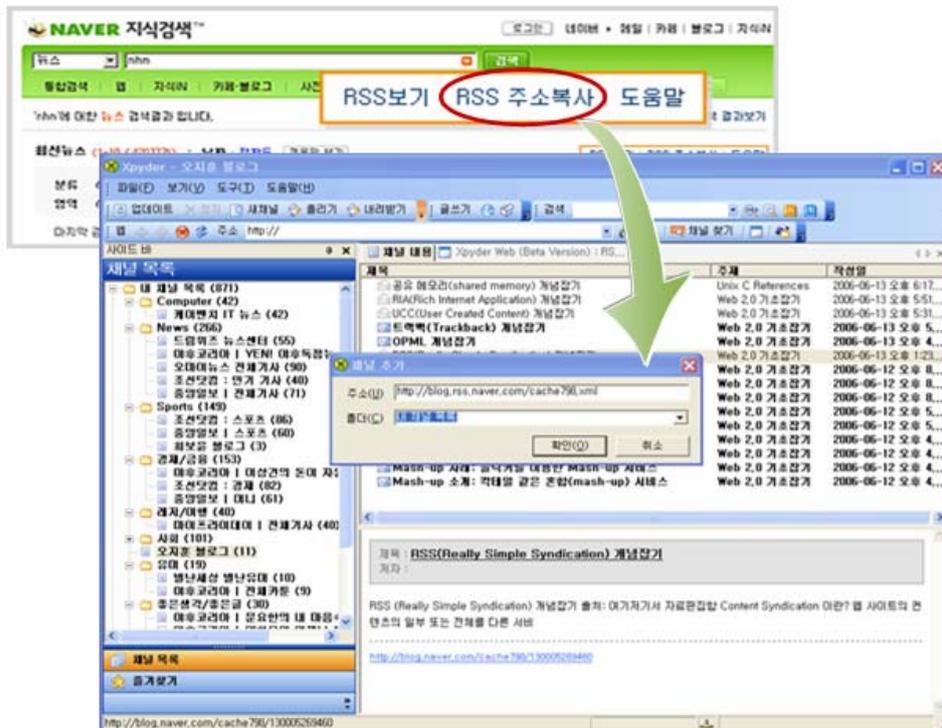
현 기술관점에서 좀더 풍부한 사용자 경험을 제공하고 좀더 혁신적인 기술이 사용된 기술을 RIA로 본다면, HTML 페이지로 웹 사이트가 정적으로 구성되었던 때에 자바스크립트, CSS 등장해서 좀더 풍부한 UI 제공하고, 거기에 Java 애플릿도 사용하였다면, 이것도 기존 환경에 비해 RIA가 적용된 기술이라고 할 수 있지 않을까?

2.3 RSS (Really Simple Syndication)

웹 사이트의 컨텐츠의 일부 또는 전체를 다른 서비스에서 이용 가능하게 해주는 것으로 Syndicated Content (Feed)는 컨텐츠 자체와 메타 데이터로 구성된다. RSS는 다양한 웹 사이트 상의 컨텐츠를 요약하고, 상호 공유하고, 주고 받을 수 있도록 만든 표준이다.

- RSS는 웹사이트 상의 컨텐츠를 요약하고, 상호 공유하고, 주고 받을 수 있는 표준으로 뉴스나 블로그와 같이 컨텐츠 업데이트가 빈번한 웹사이트에서, 업데이트된 정보를 쉽게 사용자들에게 제공하기 위해 XML을 기초로 만들어진 데이터 형식이다.
- 사이트가 제공하는 RSS 주소를 RSS Reader 프로그램에 등록하면, 업데이트된 정보를 찾기 위해 사이트에 매번 방문할 필요 없이 손 쉽게 이들을 이용할 수 있다.
- RSS 주요 사용분야
 - 알림: 매시간 새로운 정보가 추가,변경 되는 뉴스, 신규소식 서비스
 - 강좌: 고객이 매번 사이트를 방문하여 규칙적으로 확인하지 않는 컨텐츠 서비스
 - 일정: 주요 행사, 마감일자 또는 휴일정보
 - 이외에 입찰정보, 채용정보

다음 그림은 국산 RSS Reader 인 Xpyder Reader를 이용한 RSS를 이용하는 화면이다.



(그림) Xpyder Reader를 이용한 RSS 사용하기

지금까지 인터넷 이용자는 정보에 접근하기 서핑을 하다가 일반적으로 어느 사이트가 맘에 들 경우 북마크에 저장하였다. 이러한 방식은 직접 방문하지 않고서는 해당 사이트가 업데이트가 되었는지, 새 글이 올라왔는지 알 수가 없다. 하지만 RSS를 이용하면 직접 방문하지 않고서도 RSS Reader를 이용하여 사이트 업데이트 유무를 쉽게 확인할 수가 있다.

다음은 RSS 샘플 소스이다.

```
<?xml version="1.0"?>
<rss version="2.0">
  <channel>
    <title>Example Channel</title>
    <link>http://example.com/</link>
    <description>My example channel</description>
    <item>
      <title>News for September the Second</title>
      <link>http://example.com/2002/09/01</link>
      <description>other things happened today</description>
    </item>
    <item>
      <title>News for September the First</title>
      <link>http://example.com/2002/09/02</link>
    </item>
  </channel>
</rss>
```

[\(리스트\) RSS 샘플 소스](#)

2.4 OPML (Outline Processor Markup Language)

OPML과 RSS의 차이점은 블로그를 채널그룹별로 관리할 수 있다는 점이다. 즉, 여러 개의 블로그 사이트 RSS 목록을 하나의 문서 파일로 만드는 것이 OPML이다.

- OPML은 '개요 처리 언어'이다. 이런 면에서 보면 홈페이지의 문서 내용을 요약해서 보여주는 RSS 형식과 마찬가지로 인터넷 문서 수집(신디케이팅)을 위한 문서 형식 중 하나로 볼 수 있다.
- OPML 파일은 수 백 개 블로그 사이트의 **RSS 주소를 정리한 XML 기반의 RSS 목록 파일**이다. 다른 사람의 블로그 사이트 정보를 담은 OPML 파일을 자신의 RSS 구독

기에 등록하면 다른 사람이 구독하던 양질의 블로그 사이트 목록을 공유할 수 있다.

- 자신이 구독하는 블로그 사이트 정보를 OPML 파일로 교환함으로써 블로거들끼리 양질의 블로그 사이트 정보를 공유할 수 있다. 마치 과거에 사이트 정보를 담은 북마크를 교환하면서 사이트 정보를 공유한 것과 유사하다.

다음 리스트는 OPML의 샘플 소스를 나타낸 것이다.

```
<?xml version="1.0" encoding="ISO-8859-1" ?>
<opml version="2.0">
<head>
  <title>scriptingNewsDirectory.opml</title>
  <dateCreated>Mon, 31 Oct 2005 17:23:24 GMT</dateCreated>
  <dateModified>Tue, 20 Jun 2006 17:49:17 GMT</dateModified>
  <ownerName>Dave Winer</ownerName>
  <ownerId>http://blogs.opml.org/mail/dave</ownerId>
  <expansionState />
  <vertScrollState>1</vertScrollState>
  <windowTop>163</windowTop>
  <windowLeft>626</windowLeft>
  <windowBottom>549</windowBottom>
  <windowRight>966</windowRight>
</head>
<body>
  <outline text="On this day in" created="Mon, 31 Oct 2005 18:22:29 GMT" type="link"
    url="http://archive.scripting.com/xml/onThisDayIn.opml" />
  <outline text="Scripting News Archive" created="Wed, 08 Feb 2006 21:44:03 GMT">
    <outline text="In HTML" created="Wed, 08 Feb 2006 21:44:07 GMT" type="link"
      url="http://www.scripting.com/monthlyArchiveOutline.opml" />
    <outline text="In OPML" created="Fri, 10 Feb 2006 20:24:00 GMT" type="link"
      url="http://www.scripting.com/opmlArchive/index.opml" />
  </outline>
</body>
</opml>
```

[\(리스트\) OPML 샘플소스](#)

이처럼 RSS와 마찬가지로 OPML의 활용 가치는 아직 매우 높다. OPML 수집이나 교환 사

이트가 생긴다면 좋은 RSS 지원 사이트를 서로 공유할 수 있다. 또한 사이트를 손쉽게 구별하는 등 많은 점이 편리해진다. 영화 관련 사이트만 정리한 OPML을 수집 사이트에 올리고 이를 모두 더해나간다면 손쉽게 영화 관련 사이트 목록을 만들 수 있다. 이를 통해 새로운 비즈니스 기회를 창출할 수도 있다.

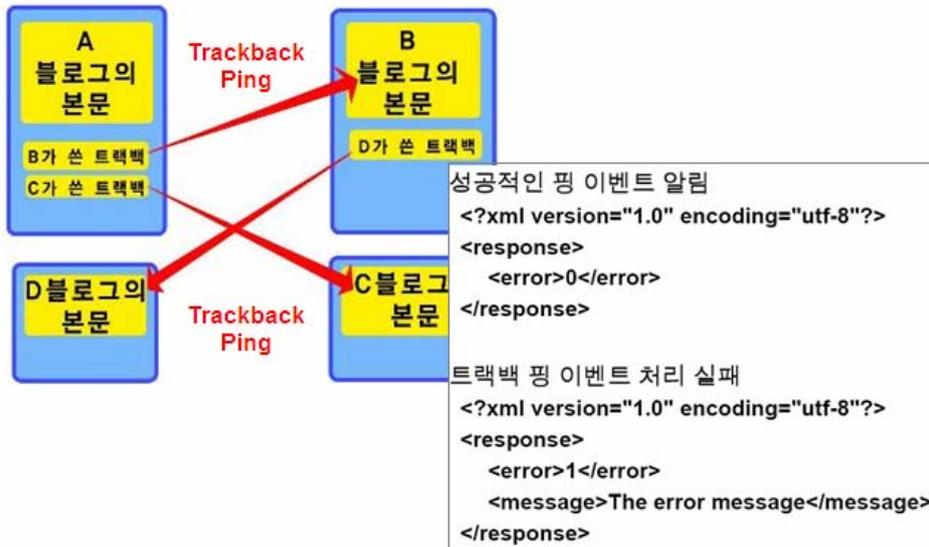
2.5 Trackback / Trackback Ping

트랙백은 다른 사람이 쓴 블로그 문서에 자신이 원격 댓글을 달았다는 사실을 알려주는 것을 뜻한다. 이는 답글(reply), 댓글(comment) 기능의 확장이다. 기존 답글과 댓글은 게시글을 읽고 난 뒤 답변을 적는 기능이어서 해당 게시물 밑에만 남겨진다.

- 트랙백은 두 블로그 사이트 사이에 연락을 취하는 수단이 됩니다. 트랙백을 통해 A는 B사이트의 글에 관심이 있다는 것을 표명할 수 있고 B는 A라는 사람이 자신이 쓴 글에 대해 관심을 가지고 있다는 사실을 알 수 있습니다.
- 트랙백의 **장점**은 콘텐츠 수집이 용이하다는 것이다. '월드컵' 글 하나를 통해 '월드컵'과 관련된 블로그 사이트들이 뭉치게 되고, 월드컵 관련 블로그 사이트들을 돌아다닐 수 있다. 하나의 글이 해당 게시판에서 게시물 하나로 끝나지 않고 관련 블로그 사이트들을 취합하는 효과를 가져올 수 있다.
- 트랙백을 건 사람은 원 게시물 작성자에게 트랙백 핑을 보내 자신의 사이트에 관련 코멘트를 달았다는 사실을 알려준다. 물론 이는 프로그램이 자동으로 보낸다.
- 트랙백의 **단점**은 Push 방식이므로 트랙백이 걸린 글은 수정하기 어렵다. A 블로거가 자신의 블로그에 트랙백으로 글을 올렸을 경우 이 글에 대한 요약문은 A 블로거 자신도 수정할 수 없다. 따라서 트랙백을 사용하는 경우에는 자신이 쓴 글의 요약문이 더 이상 수정 불가능한 글이 된다는 점을 심각하게 고려해야 한다.

트랙백으로 작성한 글은 작성자 블로그의 새 엔트리가 됩니다. 예를 들어 A 사이트의 블로거가 '한글날에 대하여'라는 글을 올렸을 경우 B 사이트의 블로거는 해당 글에 대한 의견을 자신의 블로그 사이트에 트랙백 형태로 올릴 수 있습니다.

다음 그림은 트랙백 과정을 표현한 그림이다.



(그림) 트랙백 서비스 과정

위에서 언급한 트랙백 과정을 기술하자면 다음과 같다.

1. A가 자신의 블로그에 '한글날'에 대한 글을 올렸다.
2. B가 A의 블로그에 올라간 글을 보고 자신의 블로그에 '한글날' 글에 대한 소감을 적어 글을 올렸다.
3. B는 A의 블로그에 트랙백 핑(TrackBack Ping)을 보내 자신의 블로그에 A가 쓴 '한글'에 대하여 코멘트를 달았음을 알려준다.
4. A는 자기가 쓴 '한글날' 게시물에 달린 트랙백을 통해 B가 B의 블로그에 '한글날'과 관련된 글을 올렸다는 사실을 알 수 있습니다.

2.6 Mashup Service

Mash-up 서비스는 최근에 통용되기 시작한 신조어로 여러 3rd Party의 API 또는 서비스 자원이 되는 리소스들을 혼합하여 새롭게 만들어진 웹 서비스를 의미한다. 다음 그림은 Mashup 서비스를 표현한 것이다.



(그림) Mashup 서비스 비유

Mashup 서비스의 장/단점은 다음과 같다.

- Mash-up 서비스의 장점은 새로운 서비스를 만들기 위한 비용이 매우 적다는 점이다. 기존의 공개된 자원을 활용하기 때문에 사실상 새로운 자료를 구축하기 위한 비용은 0에 가깝다. 이미 구축된 자료를 섞는 방법만 생각하고 구체화하면 된다.
- Mash-up 서비스의 단점은 다른 서비스에 종속적이라는 점이다. 1차 자원이 되는 서비스가 어느 순간 사라질지 모르기 때문에, 서비스가 중단되면 영향을 받는다. 또한 1차 자원의 제공형태가 변경될 때마다 이에 맞추어야 한다. 때문에 다양한 Mash-up 서비스가 안정적으로 제공되려면 1차 서비스의 안정적인 제공에 대한 약속과 신뢰가 필요하다.

대표적인 사례로는 구글 지도에 부동산 정보를 결합하여 지도를 선택하면 해당 지역의 부동산 매물을 보여주는 하우스징맵스를 들 수 있다.

2.7 AJAX (Asynchronous Javascript And XML)

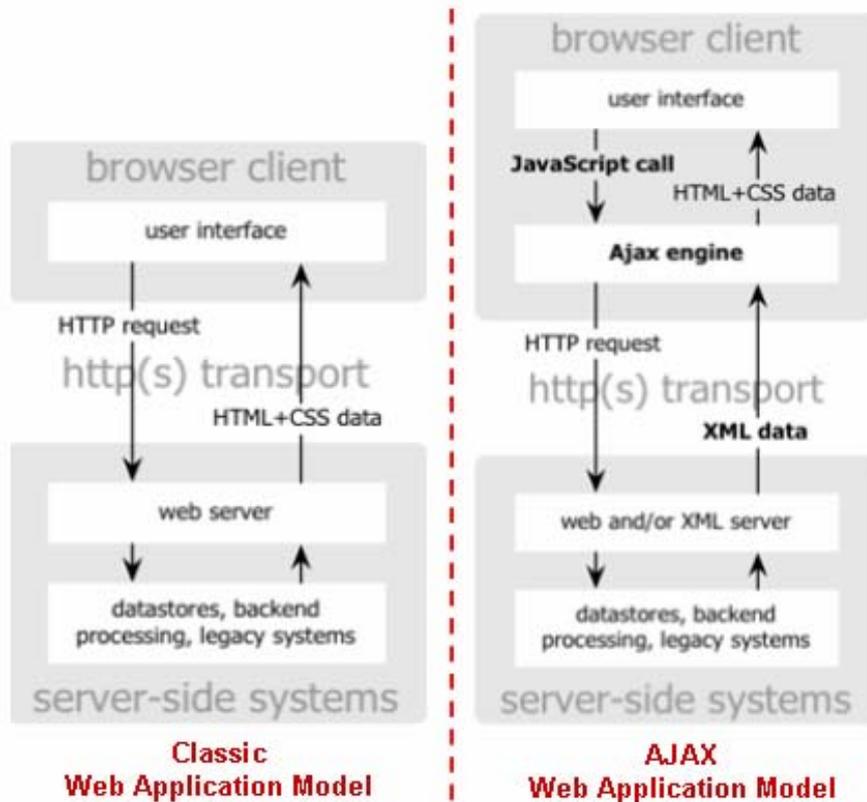
아작스는 비동기 자바스크립트 XML 이다. 이는 '브라우저와 서버 사이의 통신에는 XML를 사용하고, 사용자가 브라우저 화면의 인터페이스로는 자바스크립트 렌더링 엔진을 이용한 기술'로 개념이다.

- 아작스에 사용된 기술은 XHTML과 CSS를 사용한 표준 설계에 동적 표시, DOM을 사용한 상호작용, 브라우저와 서버간의 통신을 위한 XML, 사용자 화면의 인터페이스

스로는 자바스크립트, XMLHttpRequest를 이용한 비동기 자료 검색으로 구성된다.

- 아작스의 **단점**으로는 Javascript + XML 이라는 태생적 한계 때문에 Javascript의 언어적 한계와 구현 복잡성, 웹 브라우저 별로 상이한 DOM 핸들링 및 메모리 처리방법으로 인한 호환성 문제, Back/Forward, Bookmark등의 사용자 편의성 문제, Flash 수준의 UI API 부재 등이 있다.

다음은 기존의 웹 응용 모델과 AJAX를 이용한 웹 응용 모델을 비교한 것이다.



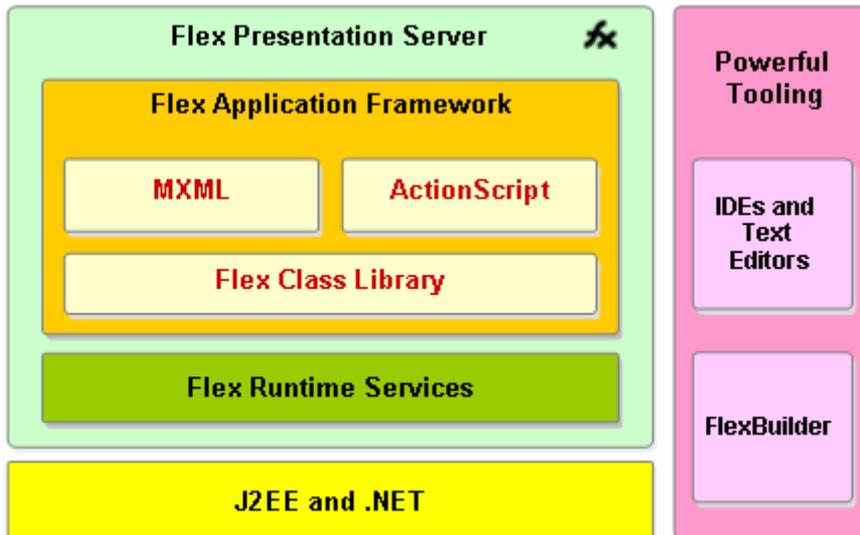
(그림) Jesse James Garrett이 비교한 이전의 모델과 AJAX 모델의 차이

대표적인 구현 사례는 Panic.com (www.panic.com), PicasaWeb (picasaweb.google.com), 구글 개인화 사이트 (www.google.com/ig) 등이 있다.

2.8 Adobe FLEX

플렉스는 디자이너들이 플래시로 개발하는 화면을 개발자들이 XML(MXML) 및 스크립트 언어로 화면을 개발할 수 있도록 만든 기술로, 이는 데스크탑의 범용성과 웹의 광범위함을 접목시킨 RIA의 UI를 구현 가능한 최적의 솔루션이다.

다음 그림은 플렉스 서버의 구조이다.

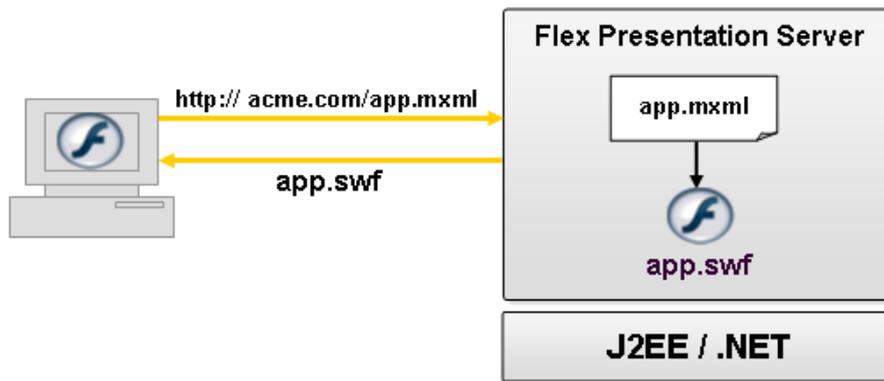


(그림) Flex 서버 구조, <http://www.adobe.com/kr/products/flex/>

다음은 플렉스의 기본사항을 설명한 것이다.

- 플렉스의 mxml은 'mx'라는 XML 네임스페이스를 사용하며 XML 문법을 따르며, ECMAScript는 플렉스 액션스크립트가 준수하는 표준으로 자바스크립트와 유사하다. 또한 MXML 스타일은 CSS 문법을 지원하고, MXML 파일은 UTF-8로 작성 및 저장되어 서버에서 처리된다.
- 사용자가 플렉스로 구현된 사이트로 접속하면 웹애플리케이션 서버에 설치된 플렉스 서버는 MXML 코드를 SWF(플래시 실행파일)로 컴파일해서 사용자 PC에 전송되어 플래시 플레이어에 이를 보여준다. 사용자의 플래시플레이어에서 구동되기 때문에 사용자는 브라우저에 상관없이 동일한 화면을 보여준다.
- Ajax의 여러 가지 문제점을 극복하고 플래시 수준의 화면을 보여주는 RIA를 만들어 Web 2.0을 실현하는 최선의 대안이 플렉스이다.

다음 그림은 플렉스 서비스 절차를 나타낸 것이다.



(그림) 플렉스 서비스 절차

다음은 MXML(Macromedia Flex Markup Language)과 SWF 파일(app.mxml, app.swf)를 나타낸 그림이다.

```

<?xml version="1.0"?>
<mx:Application xmlns:mx="http://www.macromedia.com/2003/mxml" xmlns="*"
  <!-- calculator controller -->
  <CalculatorHandlers id="calcHandlers" calcView="(this)"/>
  <!-- calculator view -->
  <mx:Panel title="Calculator">
    <!-- calculator display -->
    <mx:Label id="calcDisplay" width="150" textAlign="right"/>
    <!-- calculator controls -->
    <mx:Grid>
      <mx:GridRow>
        <mx:GridItem colSpan="2">
          <mx:Button width="70" label="Clear" click="calcHandlers.clear"/>
        </mx:GridItem>
        <mx:GridItem>
          <mx:Button width="30" label="C/E" click="calcHandlers.clear"/>
        </mx:GridItem>
        <mx:GridItem>
          <mx:Button width="30" label="+" click="calcHandlers.add"/>
        </mx:GridItem>
      </mx:GridRow>
      <mx:GridRow>
        <mx:GridItem>
          <mx:Button width="30" label="1" click="calcHandlers.number1"/>
        </mx:GridItem>
        <mx:GridItem>
          <mx:Button width="30" label="2" click="calcHandlers.number2"/>
        </mx:GridItem>
        <mx:GridItem>
          <mx:Button width="30" label="3" click="calcHandlers.number3"/>
        </mx:GridItem>
        <mx:GridItem>
          <mx:Button width="30" label="-" click="calcHandlers.subtract"/>
        </mx:GridItem>
      </mx:GridRow>
      <mx:GridRow>
        <mx:GridItem>
          <mx:Button width="30" label="4" click="calcHandlers.number4"/>
        </mx:GridItem>
        <mx:GridItem>
          <mx:Button width="30" label="5" click="calcHandlers.number5"/>
        </mx:GridItem>
        <mx:GridItem>
          <mx:Button width="30" label="6" click="calcHandlers.number6"/>
        </mx:GridItem>
        <mx:GridItem>
          <mx:Button width="30" label="*" click="calcHandlers.multiply"/>
        </mx:GridItem>
      </mx:GridRow>
      <mx:GridRow>
        <mx:GridItem>
          <mx:Button width="30" label="7" click="calcHandlers.number7"/>
        </mx:GridItem>
        <mx:GridItem>
          <mx:Button width="30" label="8" click="calcHandlers.number8"/>
        </mx:GridItem>
        <mx:GridItem>
          <mx:Button width="30" label="9" click="calcHandlers.number9"/>
        </mx:GridItem>
        <mx:GridItem>
          <mx:Button width="30" label="/" click="calcHandlers.divide"/>
        </mx:GridItem>
      </mx:GridRow>
      <mx:GridRow>
        <mx:GridItem>
          <mx:Button width="30" label="0" click="calcHandlers.number0"/>
        </mx:GridItem>
        <mx:GridItem>
          <mx:Button width="30" label="." click="calcHandlers.decimal"/>
        </mx:GridItem>
        <mx:GridItem>
          <mx:Button width="30" label="=" click="calcHandlers.equals"/>
        </mx:GridItem>
      </mx:GridRow>
    </mx:Grid>
  </mx:Panel>
  </mx:Application>

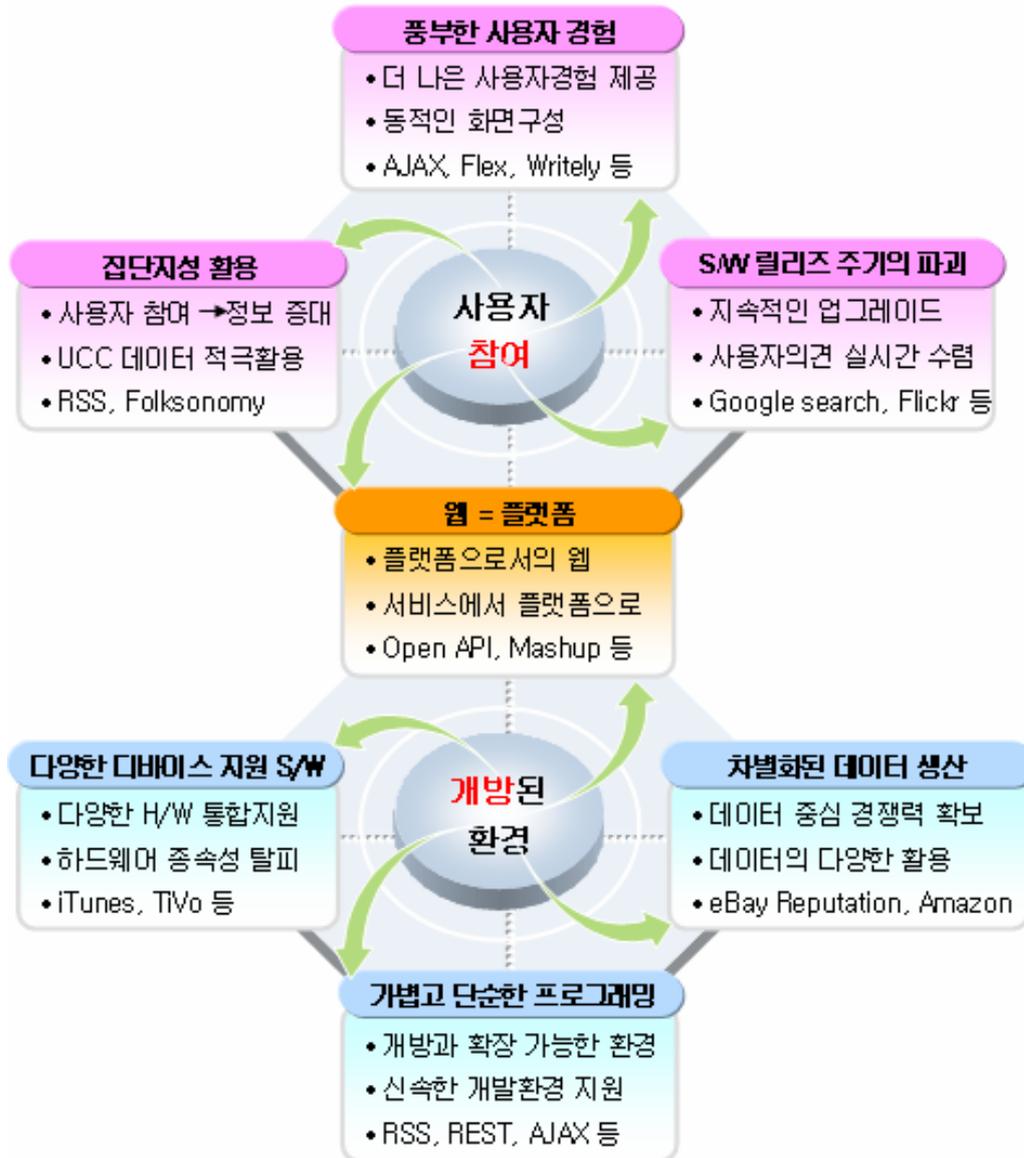
```

The image shows the MXML code for a calculator application and a screenshot of the application running in a browser. The MXML code defines the application structure, including the calculator controller, handlers, and view. The view consists of a panel titled "Calculator" with a display area and a grid of buttons for clearing, calculation, and digit entry. The screenshot shows the calculator interface with the number 33 displayed.

(그림) MXML과 SWF 파일

3. Web 2.0 특징

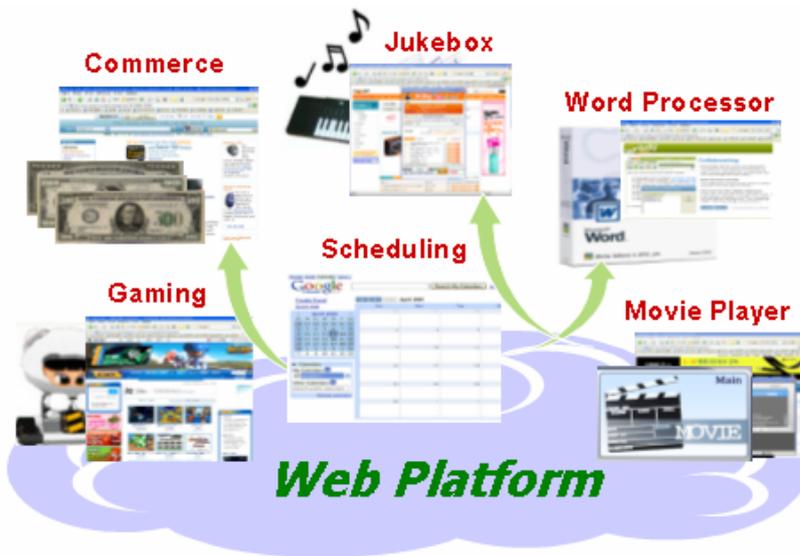
이 장에서는 Web 2.0의 7가지 특징, 즉 살아남은 닷컴 기업들의 7가지 특징에 대해 간략히 기술한다. 다음 그림을 먼저 살펴본 후 각 항목별로 자세한 내용을 확인하자.



(그림) Web 2.0 특징 요약도

3.1 웹을 플랫폼으로 생각한다.

Web 2.0은 개발자들이 만드는 소프트웨어 같은 기술적인 이슈가 아니라 웹을 쓰는 사람들이 직접 생활, 미디어, 비즈니스, 커뮤니티를 위해 쉽게 만들고 공유하고 관리할 수 있는 '플랫폼으로서 웹'이다. 다음은 이를 표현한 그림이다.



(그림) Web 2.0 플랫폼

- 웹사이트에 링크하여 기능 사용을 사용하던 차원을 넘어 필요한 기능 요소를 가져다 다른 서비스에 내장하여 사용하여 웹을 서비스 차원에서 플랫폼 차원으로 활용.
- Google 또는 Amazon과 같은 기업은 패키지 소프트웨어를 개발하고 판매하는 예전 방식을 버리고, 웹을 플랫폼으로 활용하여 서비스로서 소프트웨어를 제공하는 방식을 채택.
- 현재는 플랫폼 대 플랫폼의 전투로서 MS와 같은 기존의 인스톨 기반의 소프트웨어 플랫폼과 커뮤니케이션을 지향하며 개방형 표준과 협력에 기반한 웹과 같은 플랫폼 중 어느 것이 미래의 비즈니스 모델에 더욱 적합한 것인지 경쟁하고 있다.

대표적인 사례로는 공개 API, 다양한 Mashup 서비스, Google AdSense 등이 있다.

3.2 집단 지성을 활용한다.

Web 2.0은 더 많은 사람들이 사용할 수록 서비스는 자동으로 우수해 지고, 지능형 브로커 처럼 서비스가 동작하고, 사용자들을 상호 연결시키며, 사용자들의 힘을 이용하는 “사용자 참여의 아키텍처”이다.

- 사용자들이 개별적으로 가지고 있는 사진, 북마크, 지식 등의 컨텐츠들을 제공할 수 있도록 유도하고, 이들을 지속적으로 수집하여 거대한 데이터베이스를 형성한다.
- 사용자들은 자신들의 참여를 통하여 데이터베이스가 형성되어 가는 과정을 직접 확

인할 수 있으므로 참여도나 충실도가 높아지게 되며, 사용자들이 콘텐츠의 제공자이자 활용자가 되기 때문에 집단 커뮤니티의 결속력 강화한다.

- Web 2.0은 참여 아키텍처를 강조하고 있으며, 사용자의 가치를 중요시 하고 있어, Web 2.0 기업의 성공의 열쇠는 어떤 전략과 정책으로 사용자의 참여를 유도하고 네트워크 효과를 낼 것인가에 있다.

대표적인 사례는 del.icio.us, BitTorrent, Blog, RSS, Wikipedia 등이 있다.

3.3 차별화된 데이터로 승부한다.

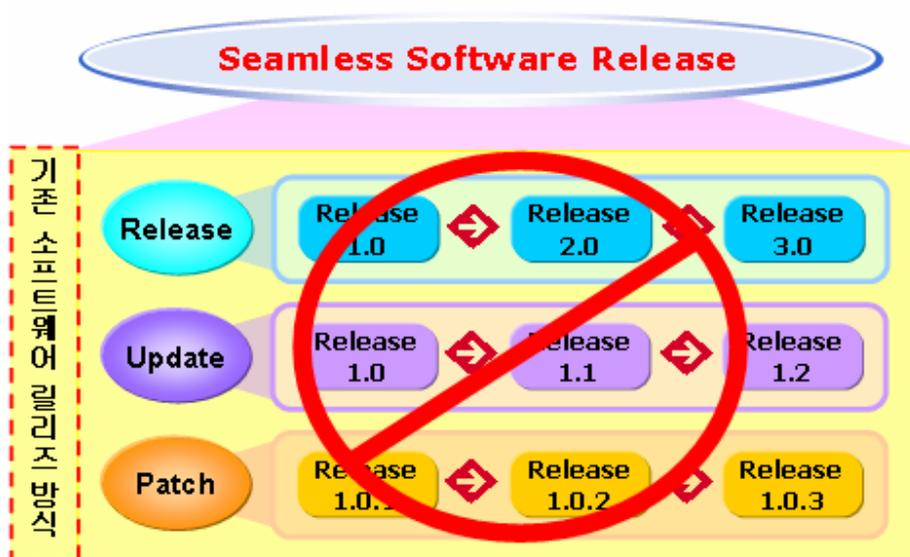
Web 2.0 시대에는 데이터를 단순히 이용하는데 그치지 않고, DB를 지속적으로 관리 및 갱신함으로써 데이터의 경쟁력 강화, 질의 향상을 통해 데이터 차별화로 시장 우위를 선점할 수 있다.

- 특정 비즈니스 영역에서 주도권을 확보하고 지속시키는 것은 그 기업이 현재 어떤 데이터를 가지고 있으며, 어떻게 관리해 나가느냐에 의해서 결정된다.
- 데이터 중심의 경쟁력 확보를 위해 단일화된 데이터 유통 채널을 개선하여 데이터 자체를 다양하게 활용하여 표준화된 데이터 유통을 통한 수익을 창출한다.
- 향후에는 “Intel Inside”의 마크가 주었던 효과처럼 어떤 데이터에 기반한 서비스가냐가 고객 선택에 차별성을 주게 될 것이다.
- 응용 프로그램은 점차 데이터 중심으로 변화하고 있기 때문에, 누구도 따라올 수 없으며 유일무이하면서 재생산하기 어려운 데이터 소스를 소유할 방안을 강구한다.

대표적인 사례는 eBay 평판 시스템, Google AdSense, Amazon 상품평 등이 있다.

3.4 소프트웨어 릴리스 주기의 혁신을 꾀한다.

Web 2.0에서는 웹 서비스가 고정된 정보를 전달하는 것이 아니라 지속적으로 자신의 DB를 갱신하기 때문에, 어제는 검색되지 않던 사이트가 오늘은 검색되는 것과 같이 서비스로서의 소프트웨어는 일일 단위로 유지 보수되고 업데이트된다.



(그림) Web 2.0 소프트웨어 릴리즈

- 기존의 클라이언트 기반 소프트웨어는 길고 정기적인 업그레이드 정책으로 인하여 빠른 업그레이드가 어려웠으나, 웹 중심의 응용 프로그램들은 빠르고 지속적인 업그레이드가 가능하다. 이로 인해 사용자 의견 및 반응에 대한 실시간 피드백 및 사용자 불만을 최소화하여 수익을 극대화한다.
- 디바이스와 프로그램이 웹에 연결되면서 응용 프로그램은 더 이상 소프트웨어 상품이 아닌 서비스로서 제공된다. 따라서 일반 사용자 경험을 토대로 지속적으로 기능을 추가하여, 사용자를 실시간 테스터로 끌어들여 사용자 의견을 최대한 반영.
- Web 2.0에서는 사용자를 신뢰하여 사용자들을 공동개발자로 간주한다. 웹사이트 개발자는 새로운 기능을 사이트에 추가하며, 사용자 행동을 실시간 체크하여 해당 기능이 사용되지 않으면 삭제하고, 반응이 좋으면 사이트 전체로 확대한다.

대표적인 사례는 Google Search, Firefox, Flickr 이미지 검색 등이 있다.

3.5 가벼운 프로그래밍 모델을 사용한다.

Web 2.0은 궁극적으로 가벼운 사용자 인터페이스(Ajax), 가벼운 개발 기술, 가벼운 프로그래밍 모델을 사용하며, 이상적인 디자인보다는 단순한 실용주의를 추구한다.

- Web 2.0은 Loosely coupled 시스템을 위한 가볍고 단순한 프로그래밍 모델을 지원한다. 이는 신속하고 지속적인 업그레이드와 개발지원뿐만 아니라 개방과 확장이 용이한 비즈니스 모델을 제시함으로써 외향적인 디자인 보다는 단순한 실용주의 추구

를 통한 사용자 편의와 서비스의 제사용을 극대화할 수 있다.

- Amazon 웹서비스는 2가지 방식으로 제공되고 있는데, SOAP을 이용한 엄격한 구조의 웹서비스 보다는 HTTP를 이용하여 XML을 주고받는 간단한 웹서비스, REST 방식의 이용률이 훨씬 높은 것으로 나타났다.(REST 방식 95%)
- 메이저 업체들의 복잡한 웹서비스 스택들은 아직도 널리 퍼지지 못하고 있는 것과 달리 RSS는 그 자체의 단순성 때문에 가장 널리 배포된 단일 웹서비스이다.

대표적인 사례는 RSS를 이용한 Content Syndication, Ajax를 이용한 사용자 화면 설계, REST를 이용한 서비스 제공 등이 있다.

3.6 단일 디바이스 수준을 넘어선 소프트웨어를 지향한다.

Web 2.0에서는 응용 프로그램을 설계부터 휴대 단말, PC, 인터넷 서버 등의 다양한 디바이스를 지원을 타겟으로 설계한다.



(그림) TiVo & iPod + iTunes

- 인터넷 애플리케이션을 접근하여 사용할 수 있는 디바이스는 PC뿐만이 아니므로, PC와 같은 특정 디바이스로 밖에 이용할 수 없는 애플리케이션은 디바이스에 무관

하게 이용할 수 있는 것보다 가치가 없다.

- 다양한 하드웨어에 대한 통합적인 지원을 통해 하드웨어-소프트웨어-서비스-데이터 간의 유연한 협업이 가능하다.

대표적인 사례는 iPod + iTunes, TiVo 등이 있다.

3.7 풍부한 사용자 경험을 제공한다.

Web 2.0에서는 웹을 단순한 컨텐츠나 정보 제공의 역할로 국한시키지 않고, 다양한 응용 프로그램 혹은 서비스가 다양한 디바이스를 통해서 전달되는 풍부한 사용자 경험의 마켓으로 활용할 수 있다.

- 단순한 HTML 페이지 링크 체계에서 한 페이지에서 동적인 화면 구성으로 전환하여 더 나은 사용자 경험을 제공하고, 웹을 통해서도 PC에 필적하는 풍부한 사용자 인터페이스와 양방향성을 갖춘 응용 프로그램을 제공한다.
- 라이틀리와 같은 Web 2.0 워드 프로세서는 통상적인 문서편집기의 기능뿐만 아니라 위키 스타일의 협업적인 문서편집기능 등이 추가되어 있다.

대표적인 사례로는 RIA, AJAX, Flex, Writely 등이 있다.

4. Web 2.0 사례 사이트 분석

이 장에서는 Web 2.0 개념을 적용한 사례 사이트에 대해 간단히 기술한다.

4.1 del.icio.us (del.icio.us), 딜리셔스

키워드: 폭소노미(folksnomy), 태그(tag), 태깅(tagging), 북마크(bookmark), 북마크공유, 북마크검색, 사용자참여 등.

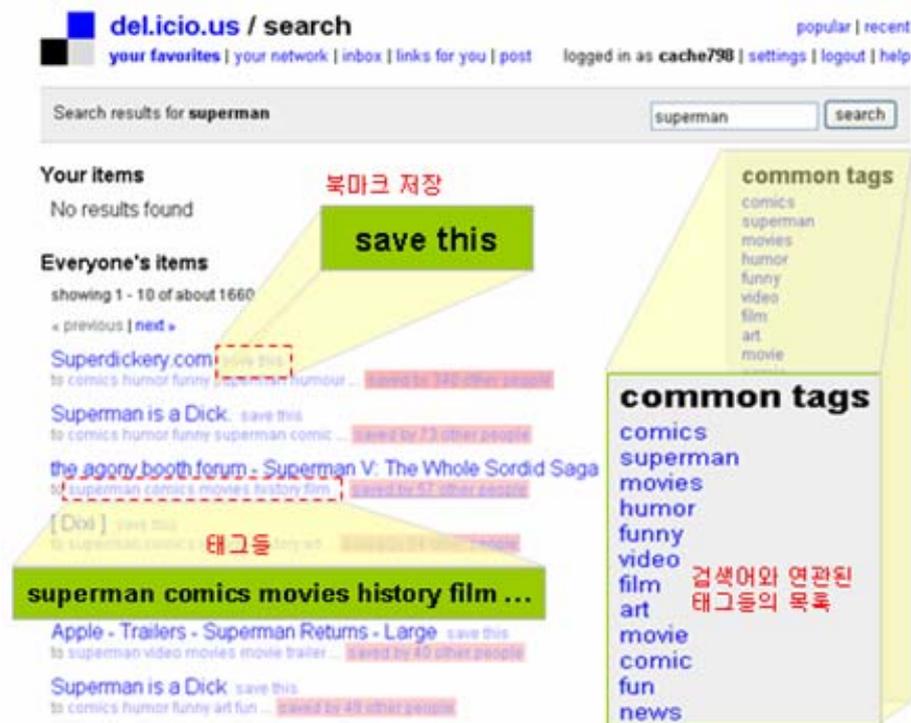
온라인에서 북마크를 저장하여 공유할 수 있는 서비스로써 태깅(tagging)을 사용하여 북마크를 저장 및 분류하여 사용자간의 북마크를 공유하는 사이트이다.

딜리셔스의 대표적인 특징은 다음과 같다.

- 폭소노미(folksnomy) 방식의 분류체계를 적용하였으며, 사이트 등록시 태그(tag)를 달아서 등록하므로 주제에 따라 검색이 용이하다.
- 폭소노미가 기존의 분류체계와 다른점은 구성원들이 자발적으로 개별정보에 의미를 부여함으로써 단위정보를 체계화한다는 것이다
- 등록된 사이트의 북마크 횟수를 집계하여 현재 가장 인기있는 사이트를 보여줄 뿐만 아니라 시간 단위로 등록된 사이트들도 보여준다.
- 사이트 검색시 입력된 검색어와 연관된 검색어를 제시하여 준다.
(검색어: game, 검색어 관련단어: gaming, fun, puzzle, online, programming, ...)
- RSS를 지원하므로 새롭게 업데이트되는 북마크를 실시간으로 제공 받아볼 수 있다.
- 현재 야후에 인수된 상태이다.



(그림) del.icio.us 시간대별 인기사이트 표시화면



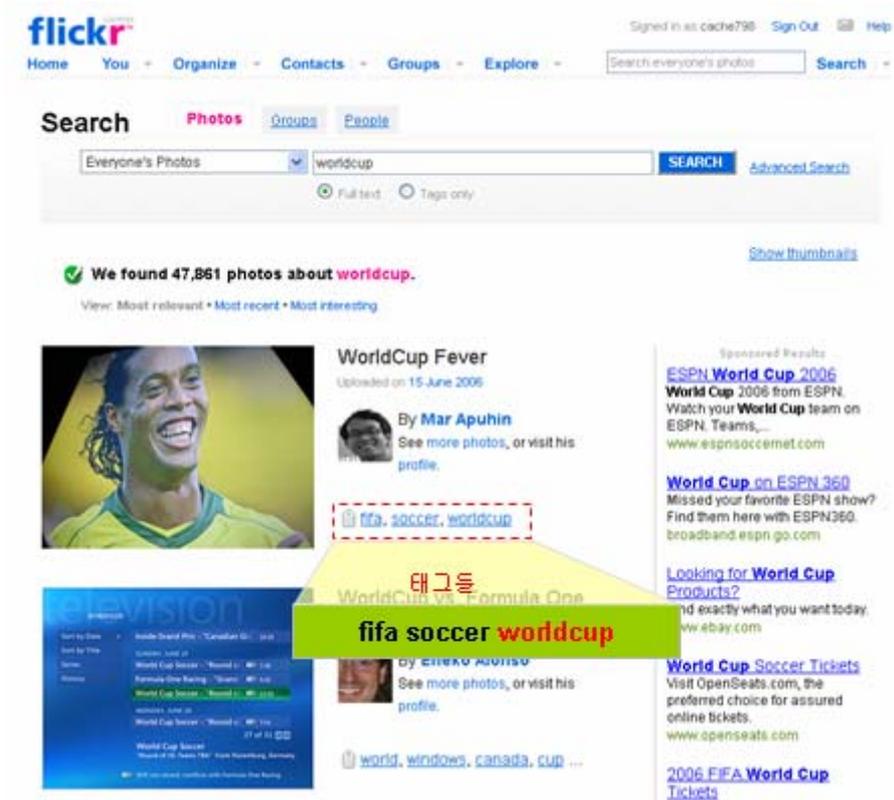
(그림) del.icio.us 사이트(북마크) 검색화면

4.2 Flickr (www.flickr.com), 플리커

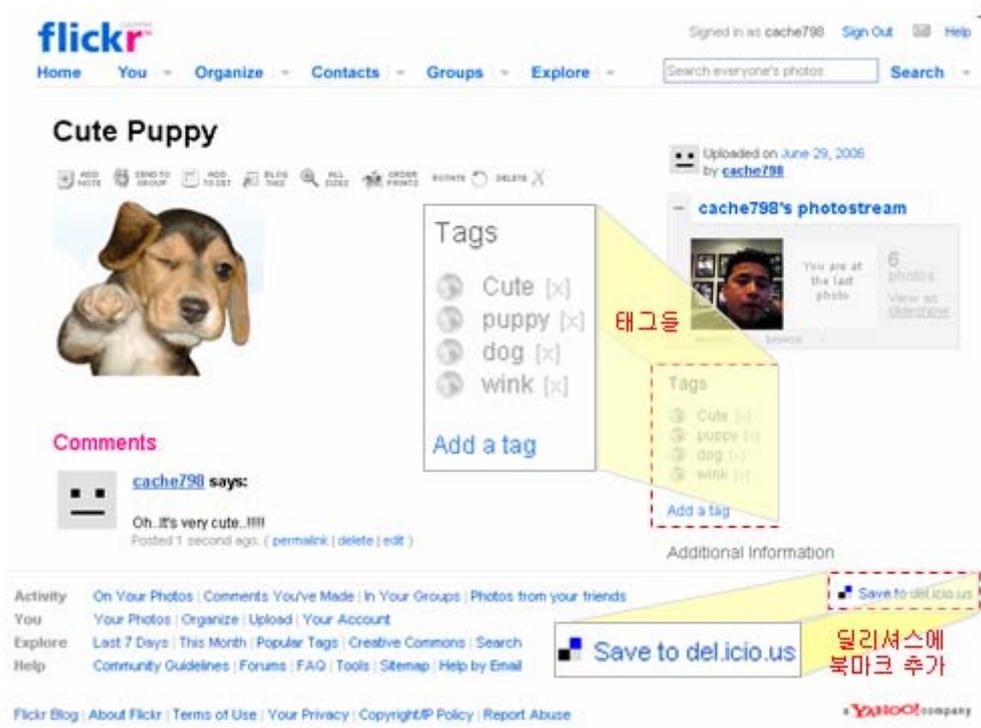
키워드: 태그(tag), 태깅(tagging), 폭소노미(Folksonomy), 사진검색, 사진공유, 사용자참여 등.

플리커는 사진 등록시에 사용자가 직접 사진 관련 태그를 입력함으로써 웹 상에서 사진 저장, 분류, 검색, 관리를 용이하게 할 수 있는 신개념의 사진 전문 분류 및 공유사이트이다. 플릭알의 대표적인 특징은 다음과 같다.

- 일반사진 검색, 사진그룹 검색, 사진관련 인물검색등 다양한 검색 기능을 지원하며, 사진 업로드할 때 사용자가 직접 태그를 부여할 수 있는 기능을 제공한다.
- 사용자가 업로드한 사진들을 손쉽게 분류할 수 있는 기능을 지원한다. 예를 들면, 제목을 붙이고, 사진에 대한 코멘트를 달고, 사진과 관련된 태그들을 달 수 있는 기능을 제공함으로써 키워드별로 사진을 분류 및 정리할 수 있다.
- 플리커는 사진을 올린이와 해당 사진에 관심을 가지고 있는 사람들과의 사진에 대한 의견을 공유할 수 있는 Discuss 기능을 지원한다.
- Del.icio.us와 연동 기능이 제공되므로 사진들을 북마킹을 통하여 del.icio.us를 통해 관리 및 공유가 가능하다.
- RSS를 지원하므로 새롭게 업데이트 되는 사진을 확인할 수 있다.
- 유사 사이트로는 구글에 인수된 Picasa(picasaweb.google.com) 등이 있다.
- 현재 야후에 인수된 상태이다.



(그림) 플리커를 통한 사진검색



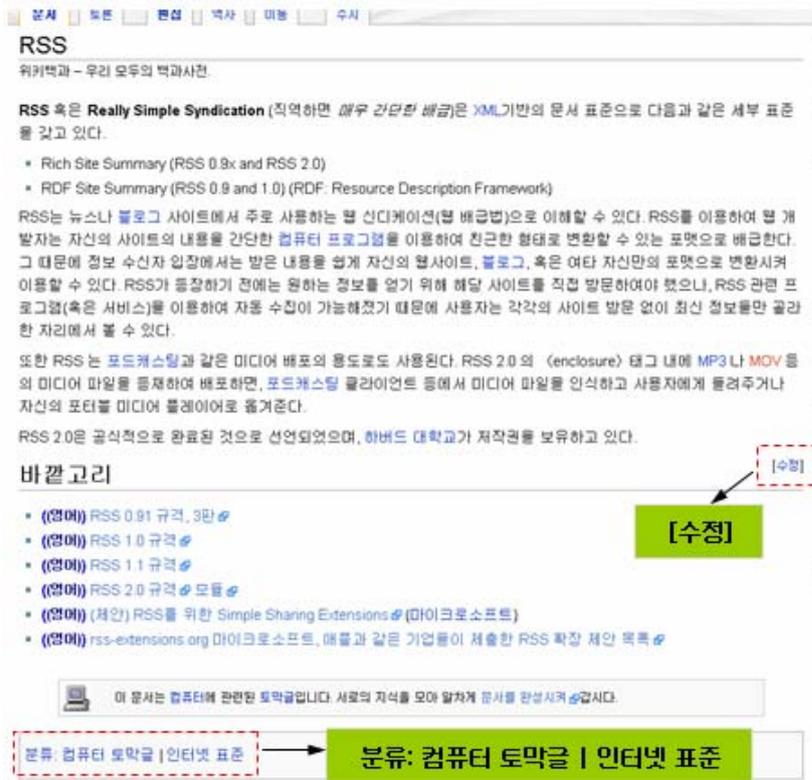
(그림) 등록된 사진 정보확인

4.3 Wikipedia (www.wikipedia.org), 위키피디아

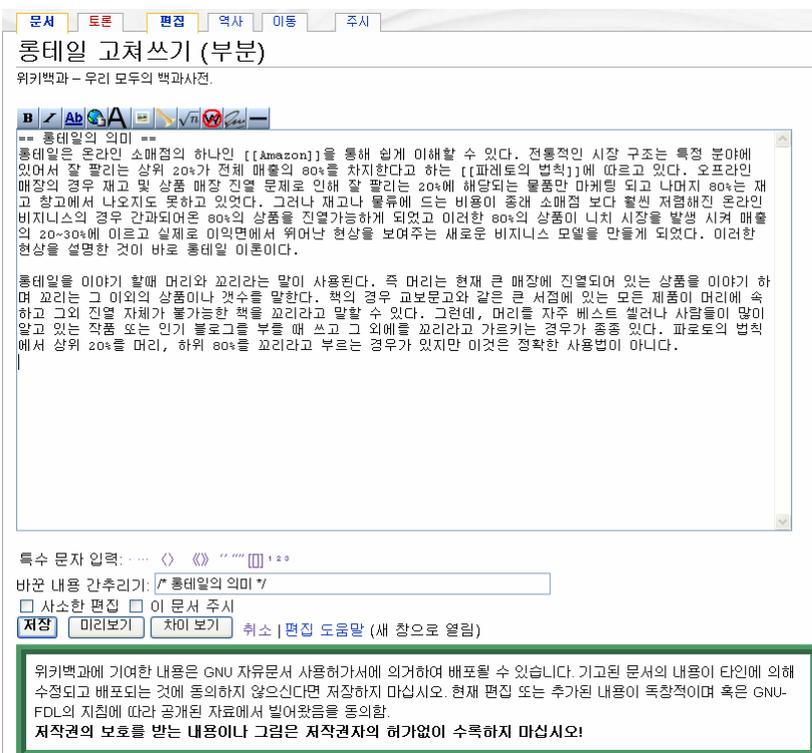
키워드: 온라인 백과사전, 위키, 사용자참여, 집단지성 등.

위키피디아는 사용자 참여에 의해 운용되며, 누구나 자유롭게 사용할 수 있는 온라인 백과사전이다. 위키피디아의 대표적인 특징은 다음과 같다.

- '위키'로 만든 백과사전(encyclopedia)이라는 의미로 배타적인 저작권이기 때문에 사용 제약을 받지 않는다. [참고] 위키피디아 한국어 사이트(ko.wikipedia.org)
- 사용자가 직접 참여하여 콘텐츠를 제공하거나 수정할 수 있는 온라인 백과사전으로 온라인 최대의 백과사전이다. 한국어를 비롯한 다양한 언어를 지원한다.
- 위키피디아는 이미 세계 최고로 꼽는 브리태니커 백과사전의 위상을 위협할 만큼 성장하였으며, 항목 수나 조회 수에서는 비교가 안 될 정도로 내용이나 공신력에서도 브리태니커에 결코 뒤떨어지지 않는다.
- 감수를 담당하는 600여명의 전문 편집자가 있으며, 별도로 상근 편집자를 두어 누구나 고쳐 쓸 수 있지만 최종적으로 내용의 정확성을 확인·검증하기 위함이다.



(그림) 위키피디어 용어 검색화면



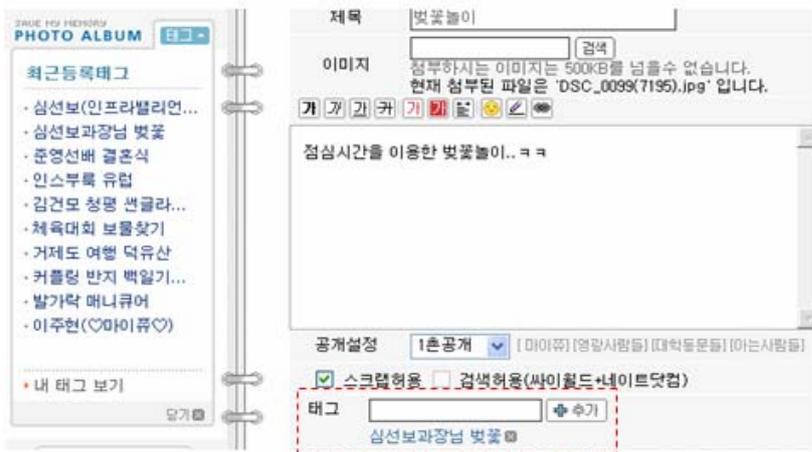
(그림) 위키피디어 사전내용 수정화면

4.4 싸이월드 태깅서비스 (www.cyworld.com)

키워드: 깡빡태그, 태그(tag), 태깅(tagging), 사용자참여, 사진검색 등.

싸이월드는 사진첨 기능에 태깅 기능을 이용하여 사진을 등록, 검색, 관리하는 태깅서비스를 지원한다. 싸이월드 태깅서비스의 특징은 다음과 같다.

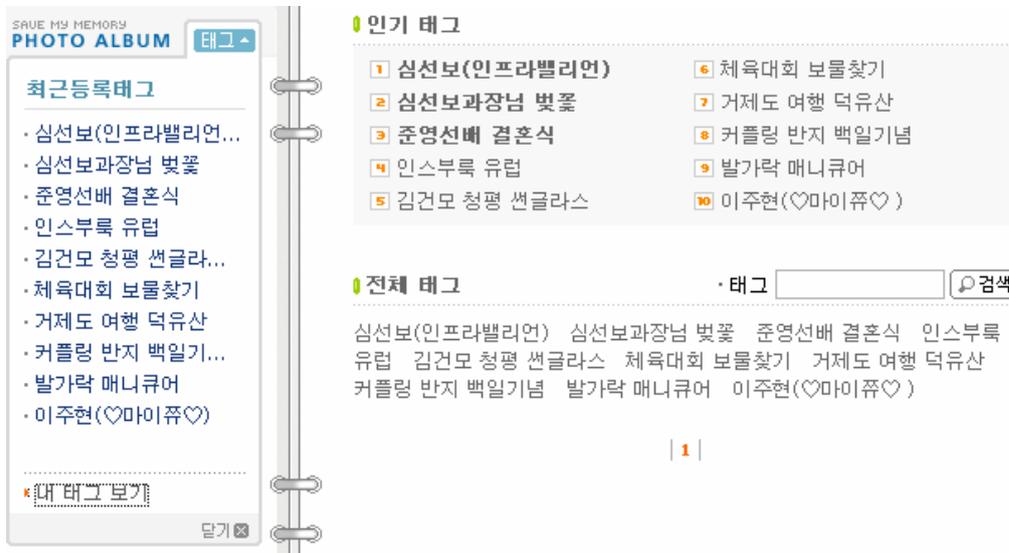
- 싸이월드에서 사용자가 입력하는 태그의 종류에는 게시물을 등록할 때 입력하는 태그와 사진 위에 입력하는 깡빡태그, 두 가지가 있다.
- 사진 아래에 있는 메뉴에서 깡빡태그를 클릭하면 사진 위에 깡빡태그를 입력할 수 있는 폼이 생긴다. 이때 사각형의 영역을 얼굴에 맞게 크기를 조정하고 이동시킨 후 사진 속 주인공의 이름을 일촌목록 중에서 선택해 깡빡태그를 입력할 수 있다.
- 사진의 주제에 따라 태그를 입력해 놓았다면, 나중에 주제에 따라 쉽게 사진을 분류해볼 수 있으며, 관심있는 일촌의 사진들만 따로 모아서 볼 수도 있다. 사진첩에서 제목뿐만 아니라 태그로도 검색할 수가 있어서 검색이 더 강력해졌다.
- 내 일촌 중 누군가가 깡빡태그에 내 이름을 입력했다면 깡빡태그 알림을 통해 확인할 수 있다. 깡빡태그 알림은 싸이월드에 로그인한 상태에서 화면 좌측에 있는 마이싸이월드를 클릭하시면 볼 수 있다.
- 깡빡태그에 붙어있는 미니홈피 아이콘을 누르면 파도타기 기능을 이용하지 않고도 바로 일촌의 미니홈피에 갈 수 있어 편리하다. 다른 사람들은 내 일촌들의 미니홈피 아이콘을 볼 수 없다.



(그림) 싸이월드 사진등록시 태그 입력화면



(그림) 싸이월드 깜짝태그 입력화면



(그림) 싸이월드 태그기반 사진검색

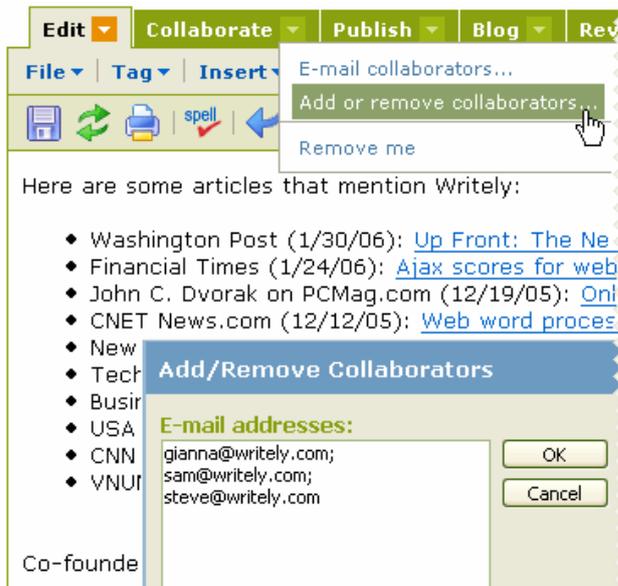
4.5 Writely (www.writely.com), 라이틀리

키워드: 온라인 문서작성, 워드프로세서, 문서공유, 문서공동작업, 문서편집, 풍부한 사용자 경험 등.

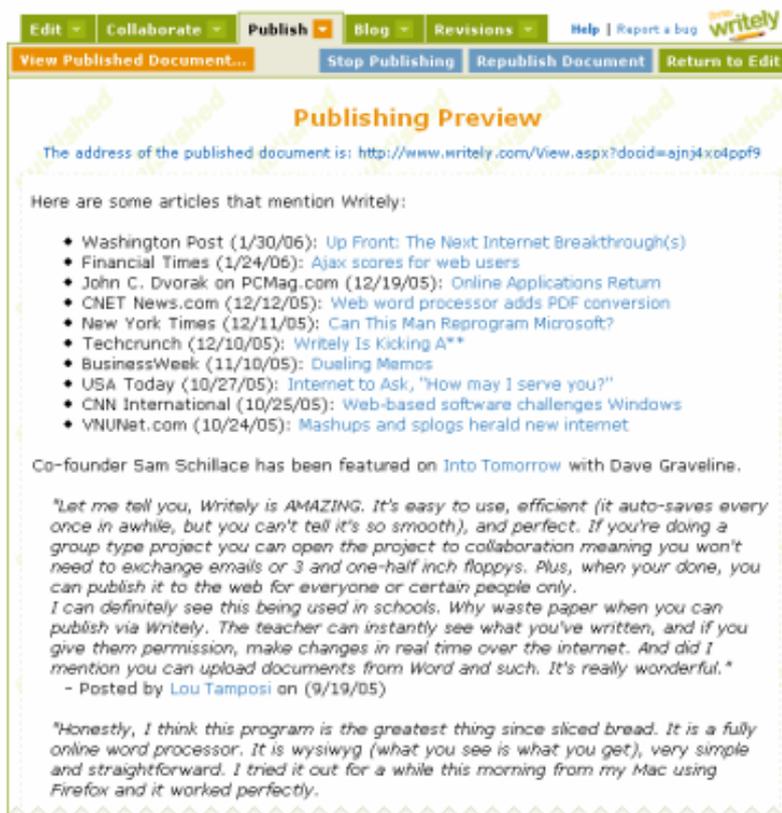
라이틀리는 웹 상에서 문서를 작성할 수 있는 온라인 워드 프로세서로써 웹 브라우저를 이용하여 어디서나 문서를 작성할 수 있는 **온라인 문서작성 사이트**이다.

라이틀리의 특징은 다음과 같다.

- 실시간으로 다른 사람들과 문서를 공유하고 공동 작성할 수 있는 기능을 제공할 뿐만 아니라 문서의 편집 기능도 지원한다.
- 문서를 버전별로 히스토리 관리하기 때문에, 다른 사람이 문서를 변경하여도 이전에 자신이 작성한 문서를 이용할 수 있다.
- 문서를 버전별로 비교하는 기능을 제공함으로써 차기 버전에서 추가된 사항을 한눈에 파악할 수 있도록 도와준다.
- 작성된 문서를 웹페이지로 변환하는 기능을 제공할 뿐만 아니라 접근 가능한 사용자들 까지도 제어할 수 있다.
- AJAX를 이용해 구현된 사이트이다.
- 라이틀리는 블로깅 기능을 지원하기 때문에 블로그의 작성, 관리, 분류가 가능하다.
- 현재 라이틀리가 구글로 통합작업 진행중이어서 사이트 신규가입이 차단되어 있다.



(그림) 라이틀리를 이용한 문서공유를 위한 작업화면



(그림) 라이틀리를 이용한 웹페이지 변환 작업화면

[사례분석] Web 2.0 특징, 개인화 체험기

키워드: 개인화, 사용자참여, 사이트 재구성 등.

4.6 Google 개인화 사이트 (www.google.com/ig)

구글 개인화 사이트는 사용자들이 자신의 의도와 구미에 맞게 사이트를 재구성할 수 있는 서비스를 제공하고 있는 사이트이다. 구글 개인화 사이트의 특징은 다음과 같다.

- AJAX 기술을 통해 사용자가 쉽게 레이아웃을 마우스 드래그만으로 변경이 가능하며, 직관적인 사용자 인터페이스를 제공한다.
- 구글 검색, Gmail, 구글검색 히스토리 기능등과 다양한 구글의 컨텐츠들 외에 RSS 형식을 지원하여 다른 사이트의 컨텐츠 업데이트 상황까지도 한눈에 파악한다.
- 구글 개인화 사이트는 좀더 사용자 친화적인 개인화 서비스를 지원하기 위해서 로그인된 사용자의 검색어 히스토리, 사이트 추가성향, 지도찾기 서비스 패턴등과 같은 정보를 수집/분석한다.
- 사용자가 자신의 성향과 관심사항, 위치 정보까지 자기도 모르게 제공함으로써 구글은 이 정보를 기초로 좀 더 고도화된 개인화 서비스나 사용자 맞춤형 광고 서비스 까지도 제공한다.
- 이럼으로써 사용자는 맞춤형 정보를 한번에 제공받을 수 있어 좋고, 구글 개인화 서비스는 사용자가 자발적으로 주는 정보를 기초로 서비스를 좀더 양적/질적으로 진화시키고 수익도 극대화할 수 있다.

The screenshot shows a Google personalization interface. At the top, there's a search bar with the Google logo and navigation links for '웹 문서', '이미지', '뉴스', '그룹스', '디렉토리', and '전체서비스'. Below the search bar are buttons for 'Google 검색' and '문 좋은 예감', along with radio buttons for '전체 웹' and '한국어 웹'. The main content area is divided into several widgets:

- Gmail**: A notification for '받은 편지함' with a link to '미리보기 숨기기' and a message from 'picasaweb-noreply' dated '6월 28일'.
- 월드컵**: A widget for the World Cup showing '다음 경기: 2006-07-01' with match details for '오전 12시 00분 - A조 - 57번째 경기' (Germany vs. Argentina) and '오전 04시 00분 - E조 - 58번째 경기' (Italy vs. Ukraine).
- 날씨**: A weather widget for '거주지 및 세계 관심 지역의 일기 예보' with a '우편번호 입력' field and a '확인' button.
- 주식 시장**: A stock market widget showing indices like DJIA (11190.80, +217.24), NASDAQ (2174.38), NYSE (8138.86), and S&P 500 (1272.87).
- del.icio.us/popular**: A list of popular links, including 'Setting Up the Perfect Ubuntu Linux Desktop' and 'Vibe Streamer - Free MP3 streaming server'.
- 검색 기록**: A list of search history items with timestamps, such as '웹 2.0' (오후 05시 07분), '인프라밸리' (오후 05시 07분), '천재' (오후 05시 07분), and '오지훈' (오후 05시 07분).
- 네이버 실시간 검색어**: A list of trending search terms from Naver, including '알렉산드라 배레산트', '빅뱅', '윤창도', '전혜빈', '몽키브라더스', '아야메일', '산별노조', '상암녀', '미키녀', and '귀상여'.

(그림) 구글 개인화 사이트 콘텐츠 배치화면

This screenshot shows the same Google personalization page but with a '나의 콘텐츠' (My Content) sidebar on the left. The sidebar contains a list of content categories with '추가 >' buttons:

- 나의 콘텐츠: 즐겨찾기, 영화
- 뉴스
- 경제
- 정보과학: Wired, Slashdot, Tech Dirt, CNET News.com
- 스포츠: Soccer Videos, ESPN, Fox Sports, Sports Illustrated, CBS Sportsline
- 라이프
- 재미
- 섹션 만들기: 주제나 피드 URL별로 검색: [검색] (예: 운동, Time Magazine 또는 engadget.com/rss.xml)

 The main content area is identical to the previous screenshot, showing the same widgets for Gmail, 월드컵, 날씨, 주식 시장, del.icio.us/popular, 검색 기록, and 네이버 실시간 검색어.

(그림) 구글 개인화 사이트 콘텐츠 추가화면

4.7 피코디 개인화 사이트 (http://www.pcodi.com/)

피코디는 의류 패션관련 포털로써 사용자가 관심있는 패션 아이템, 의류 정보뿐만 아니라 다양한 포털 콘텐츠를 조합하여 자신만의 사이트로 재구성 가능한 개인화 기능을 제공한다. 피코디 개인화 사이트의 특징은 다음과 같다.

- AJAX 기술을 통해 사용자가 쉽게 레이아웃을 마우스 드래그만으로 변경이 가능하며, 직관적인 사용자 인터페이스를 제공한다.
- 사용자 정의 Tab(탭) 기능을 지원해 주어 콘텐츠별 분류가 가능하다.



(그림) 피코디 콘텐츠 선택화면



(그림) 피코디 콘텐츠 배치화면

5. 맺음말

이 장에서는 지금까지 살펴본 Web 2.0에 대한 총체적 결론을 제시한다.

5.1 사회적 네트워크의 형성

Web 2.0의 신디케이션과 메시징 기능은 개인과 다양한 온라인 커뮤니티를 보다 밀접하게 묶는 사회적 네트워크를 형성 가능하다.

- 위키피디아, 블로그, 네이버 지식인 등은 사용자의 참여와 지식·정보 공유를 요구함으로써 기존의 매체보다 상호작용성 수준을 높임
- 커뮤니티 내에서의 개인의 사회적 존재감이 클수록 참여하는 사용자간의 상호이해도와 친밀도가 증가되며 사회적인 행동이 촉진됨

5.2 신규 사업기회 창출

Web 2.0 출현으로 새로운 사업기회가 기하급수적으로 증가될 것으로 기대된다.

- 웹을 플랫폼으로 하고 사용자 참여에 기반하는 새로운 서비스의 제공기업이 더욱 많이 등장할 것이며, 기존의 기업들도 새로운 비즈니스 영역에 뛰어들 것으로 예상됨
- 기업 분위기 활성화: 블로그와 같은 개인 미디어를 조직 내에서 도입하면 기업 내 지식·정보 공유가 활성화되고 구성원간의 관계가 더욱 돈독해질 것임

5.3 사용자 언론 영향력 확대

블로그, 포드캐스팅(Podcasting), 인터넷 확산으로 사용자 중심의 매체의 언론 영향력이 증가할 것이다. 여기서 포드캐스팅(Podcasting)이란 인터넷을 통하여 사용자들이 오디오 파일(주로 MP3 파일)을구독할 수 있도록 함으로써, 인터넷 라디오 방송을 하는 것이나 방법을 의미한다.

- 기존의 언론이 정보수집과 전달을 주관하던 중앙집중식 방식에서 언론의 형성과 보급도 개인에 기반한 분산환경으로 변화

Appendix. Web 2.0 용어 정리

이 장에서는 Web 2.0 관련 용어에 대해 간략하게 요약 정리한다.

- AJAX (Asynchronous Javascript And XML), 아작스, 에이잭스

AJAX는 웹 프로그래밍의 한 기술이 아니라 여러 가지 기술이 복합된 방법론 또는 기술복합체이다. 이는 브라우저와 서버 사이의 통신에는 XML를 사용하고, 사용자 인터페이스는 Javascript를 이용하는 기술이다. 기존 웹 모델인 '웹 서버 ↔ 웹 브라우저' 사이에 AJAX 엔진이 중간에 위치한 '웹 서버 ↔ AJAX 엔진 ↔ 웹 브라우저' 구조가 된다.

AJAX에 사용된 기술을 보면 XHTML과 CSS를 이용한 사용자 UI 처리, DOM을 사용한 데이터 프로세싱, XML과 XSLT를 이용한 데이터 교환과 변환, XmlHttpRequest를 이용한 비동기 요청처리, 좀더 효율적인 사용자 인터페이스 지원을 위한 Javascript 등이 복합되어 있다.

- Collective Intelligence, 집단지성, 집단지능

집단지성은 개별 인간들이 만나 정보와 인식을 공유하고, 나아가 새로운 지식을 생산해 내는 것을 뜻한다. 사이버 공간에서 개개의 인간은 집단지성으로서 하나의 회로 또는 입출력 단위처럼 기능한다. 마치 개미 한 마리 한 마리가 낮은 차원의 지능을 갖지만, 더듬이를 병렬로 연결해 집단별로 최적의 행위를 찾아내는 것처럼 집단지성으로서의 인류는 이전보다 강력해진 문제 해결능력을 갖는다.

개별 네티즌들이 참여해 완성시켜 가고 있는 OS인 '리눅스'는 오늘날 집단지성이 이룬 대표적 지적 산물로 꼽힌다. Web 2.0은 이러한 다양한 경험과 노하우, 지식을 가진 이용자 다수의 참여를 통한 집단지성의 가치를 중요시한다. 집단지성이 위력을 발휘하기 위해서는 개인의 자발적인 참여가 전제되어야 한다.

- Content Syndication, 콘텐츠 신디케이션, 콘텐츠 수집

콘텐츠 신디케이션이란 사이트 콘텐츠의 일부 또는 전체를 다른 서비스에서 이용할 수 있도록 해주는 것을 말한다. 근래에 들어 특히 1인 미디어인 블로그(blog)를 중심으로 콘텐츠 신디케이션 표준인 RSS의 가능성이 확인되면서 다양한 응용들이 등장하고 있다.

- FOAF (Friend Of A Friend)

FOAF는 시맨틱 웹 기술을 적용해 관계성을 확장하는 대표적인 기술이다. '친구의 친구'라는 의미로 친구를 통해 친구를 만들어나가는 인간 관계를 말한다. FOAF는 몇 단계만 거치면 나라 안의 모든 국민을 알 수 있다는 다단계 구조를 이용하고 있다. 내가 알고 있는 1천 명

이 각기 1천 명을 알고 있다면 1단계인 친구의 친구 소개만으로도 10만 명을 내 인간관계 안에 포함시킬 수 있다.

- Folksonomy, 폭소노미

폭소노미란 사용자가 자유롭게 선택한 키워드(태그)를 통해 여러 사람의 정보를 체계화하는 분류방식이다. 흔히 사람들에 의한 분류법을 의미하는 폭소노미(Folksonomy)는 사람들을 의미하는 Folks와 분류법의 Taxonomy의 합성어이다.

폭소노미가 기존의 분류체계와 다른 점은 구성원들이 자발적으로 개별정보에 의미를 부여함으로써 단위정보를 체계화한다는 것이다. 주요 구성원과 상호작용하면서 정보가 나열되기 때문에 보다 정확하게 의미있는 정보를 전달할 수 있는 장점을 지니고 있다.

- Long Tail, 롱테일

롱테일이란 기존의 패러다임에서 사소한 것으로 간주되던 나머지 80%가 점점 더 중요해지는 것을 가리킨다. 시장의 중심이 머리에 해당하는 소수의 히트제품에서 꼬리에 해당하는 다수의 틈새 제품으로 움직여 가는 현상을 설명할 때 주로 등장하는 개념이다.

다시 말해, 롱테일은 이러한 다양한 상품과 니치에 대한 소비자 욕구가 긴 꼬리를 이루고 있으며, 실제로 이러한 니치 상품이 유의미한 매출 효과를 낳는다는 것을 의미한다.

- Mashup, 매쉬업, 혼합

Mashup 서비스란 두 가지 이상의 재료를 섞는다는 뜻이다. 웹에서는 두 가지 이상의 자원을 섞어서 새로운 자원으로 만드는 기술을 뜻한다. 예를 들어 파란의 지도 서비스에 다음의 뉴스 서비스를 섞는 혼합 서비스를 생각해볼 수 있다. 이 경우 뉴스가 발생한 지역에 깃발이 표시되어 어느 지역에서 사건이 발생되었는지 쉽게 알 수 있다.

Mashup 서비스의 장점은 새로운 서비스를 만들기 위해 들어가는 비용이 매우 적다는 점이다. 기존의 공개된 자원을 활용하기 때문에 사실상 새로운 자료를 구축하기 위한 비용이 거의 들지 않는다. 남들이 구축해놓은 자료를 섞는 방법만 생각하고 구체화시키면 된다.

- Niche Market, 니치마켓, 틈새시장

니치마켓이란 우리에게 잘 알려진 '틈새시장'이라는 말로, 남들이 쉽게 간파하는 부분을 파고들거나 거대 업체들이 투자비용이나 효율성 측면에서 파고들기 힘든 점을 간파하여 이 부분을 집중적으로 파고들어 시장을 형성하는 것을 말한다.

예를 들어, 대부분의 제품들은 소비자가 오른손잡이일 거라는 가정을 하고 만들어진다. 그

러나 대략 전체 인구의 10%정도는 왼손잡이라고 한다. 왼손잡이 제품을 전문적으로 생산하는 기업도 성공 가능성이 있다.

- OPML (Outline Processor Markup Language), 개요처리언어

OPML은 개요 처리 언어로써 홈페이지의 문서 내용을 요약해서 통지하는 RSS 형식과 마찬가지로 인터넷 문서 수집(신디케이팅)을 위한 XML 문서 형식 중 하나이다. OPML이 RSS와 다른 점은 블로그를 채널그룹(channel group=blog roll) 별로 관리할 수 있다는 점이다.

다시 말해, 여러 개의 블로그 사이트 RSS 목록을 하나의 문서 파일로 만들어 쓸 수 있는 형식이 OPML이라고 보면 됩니다. 즉, OPML 파일은 수 백 개 블로그 사이트의 RSS 주소를 정리한 RSS 목록 파일이다. 따라서 OPML 파일을 지원할 경우 블로그(외에 RSS 지원 사이트) 사이트의 RSS 주소 관리가 무척 편해진다.

또한 다른 사람이 구독하는 블로그 사이트 정보를 담은 OPML 파일을 받아서 자신의 RSS 구독 프로그램에 등록시키면 손쉽게 다른 사람이 구독하던 좋은 블로그(외에 RSS 지원) 사이트 목록을 공유할 수 있다. 이는 마치 과거에 좋은 사이트 정보를 담은 북마크를 주고받으면서 사이트 정보를 공유한 것과 유사하다.

- Personalization, 개인화

개인화(personalization)란 웹 사이트에서 사용자 개인의 특성과 기호에 맞게 페이지 화면을 편집하여 볼 수 있는 기능을 제공하는 서비스를 뜻한다. 이를 통해서 사용자가 자신의 기호, 관심, 구매 경험과 같은 정보를 웹 사이트에 제공하면 웹 사이트는 사용자가 제공한 자료를 기초로 사용자에게 가장 알맞은 정보를 선별하여 볼 수 있게 해 준다.

개인화를 통해서 웹 사이트 운영자는 사용자에게 대한 정보를 얻고 사용자의 지속적인 이용이나 구매를 얻어낼 수 있게 되며 사용자는 자신에게 가장 알맞은 정보를 편리한 방법으로 얻을 수 있다.

- RIA (Rich Internet Application)

RIA는 미국의 매크로미디어사가 2001 년 하반기 선보인 "플래시MX" 저작물을 통해 선보인 개념으로, 웹 브라우저를 기반으로 하지만 마치 윈도우 프로그램 같은 풍부한(Rich) 기능을 보이는 것을 가능하게 하는 웹 클라이언트 기술이다.

- RSS (Really Simple Syndication / Rich Site Summary)

RSS는 뉴스나 블로그 와 같이 컨텐츠 업데이트가 자주 일어나는 웹사이트에서, 업데이트된 정보를 쉽게 사용자들에게 제공하기 위해 XML을 기초로 만들어진 데이터 형식이다. 사이트

가 제공하는 RSS 주소를 자신의 RSS Reader 프로그램에 등록하면, 업데이트된 정보를 찾기 위해 사이트에 매번 방문할 필요 없이 쉽게 이들을 확인하고 이용하실 수 있다.

- Tag, 태그, 꼬리표

태그(Tag)는 특정 웹 리소스가 가지고 있는 의미를 나타내는 몇 가지 단어, 혹은 구문으로 정의한다. 카테고리는 카테고리 생성후 리소스를 추가하기 때문에 리소스를 만드는 과정에서 어떤 카테고리에 포함되어야 하는지를 고민해야 하지만, 태그의 경우는 일단 리소스를 만들어 놓고, 자유롭게 태깅을 할 수 있기 때문에 일종의 제약 조건을 고려하지 않아도 되기 때문에 좀 더 자유스러운 생성할 수 있다.

- Trackback, 트랙백, 원격댓글

트랙백을 만든 이유와 그 의미는 '내가 쓴 글을 다른 사람에게 알리기 위함'이다. 트랙백은 이를 지원하기 위한 기능이다. 즉, 트랙백은 다른 사람이 쓴 블로그 문서에 자신이 원격 댓글을 달았다는 사실을 알려주는 행위를 말한다.

트랙백을 건 다음에는 트랙백 핑(TrackBack Ping)이라고 부르는 작은 메시지를 프로그램인 자동으로 상대방에게 보내준다. 트랙백을 건 사람은 원 게시물 작성자에게 트랙백 핑을 보내 자신의 사이트에 관련 코멘트를 달았다는 사실을 알려준다.

- UCC (User Created Content), 사용자 제작 콘텐츠

UCC는 사용자에게 더 많은 의미를 두어, 이전엔 회사/사이트가 만들어 사용자에게 제공하던 것을 이전 사용자들이 만들어 공유한다 라는 의미이다. 기존에는 사이트 운영자가 제작하여 배포하던 "지식"이나 "뉴스" 같은 콘텐츠를 요즘에는 사용자들이 스스로 제작하여 공유한다.

이러한 맥락에서 UCC가 정의되고 이런 UCC를 생성하고 모아둔 플랫폼(사이트)을 Web 2.0으로 볼 수 있으며, UCC 관점에서 Web 2.0 서비스가 Web 1.0 모델과 다른 것은 사용자가 자발적으로 참여하여 UCC를 만들도록 하는 창의적인 기능이나 참여의 아키텍처를 가지고 있다는 점이다.

- Wiki, 위키

'위키위키'는 하와이 방언으로 '빨리빨리'라는 뜻인데 줄여서 '위키'라고도 한다. 위키는 보통의 웹 게시판과 달리 누구나 쓰고 고치고 지울 수 있게 돼 있다. 로그인이 필요없고 당연히 작성자의 이름도 남지 않는다. 이를테면 불특정 다수가 참여하는 공동작업 시스템인 것이다. 여기에는 여러 사람이 머리를 맞대면 더 정확한 정보가 된다는 믿음이 있다. 내가 쓴 글 가운데 잘못된 부분을 누군가가 고쳐서 다듬을 수 있고 아예 지워버릴 수도 있다.

- XFN (XHTML Friends Network)

XFN은 하이퍼링크를 이용해 인간 관계를 표현하는 간단한 수단으로 글로벌 멀티미디어 프로토콜 그룹(GMPG)에 의해 개발되었다. XFN은 단순히 <a href> 태그에 'rel' 속성을 추가함으로써, 작성자가 다른 사람들과의 관계를 표현할 수 있는 방법을 제공한다. 예를 들어 몽키몽키가 만났던 한 친구의 사이트에 대한 링크는 다음과 같이 관계를 표현할 수 있다.

```
<a href="http://www.monkeymonkey.com/" rel="friend met">몽키몽키</a>
```