특집 04

웹의 진화와 소프트웨어의 패러다임 변화

목 차

백 영 란
(한국소프트웨어진흥원)

1. 서 론

웹 2.0이 상징적으로 표현하고 있는 웹의 진화에 따라 소프트웨어도 지금 패러다임 변화를 겪고 있다. 웹의 진화하면서 진전되고 있는 소프트웨어와 서비스의 융합은 그 동안 IT산업에서 하드웨어와 소프트웨어의 결합 보다 오히려 밀착되어 있다. 우리가 현재 목격하고 있는 소프트웨어와 서비스의 융합은 두 가지 차원으로 구분되는데, 1) 소프트웨어 일반과 다양한 웹 기반 서비스, 2) 패키지 소프트웨어와 IT서비스의 융합이 그것이다. 융합에 따라 소프트웨어자체의 역할도 변화되고 있다. 웹 기반 서비스를 제공하는 수단으로써 소프트웨어의 새로운 역할을 분석하기 위해서는 웹의 진화에 따라 소비자들이 소프트웨어를 사용하는 목적을 우선 이해할 필요가 있다.

이른바 웹 2.0은 본질은 "웹을 통한 상호소통(Interaction)"이다. 소비자는 소프트웨어 그 자체가 아니라 소프트웨어를 통해 제공되는 서비스에서 사용가치 그리고 교환가치를 찾는다. 사람들의 소프트웨어에 대한 필요는 '소프트웨어를 통한 상호작용'(interaction through software)과 '소프트웨어와의 상호작용'(interaction with software)으로 구분되어진다. 사용자 사이의 상호작용을 목적으로 하는 웹 2.0 프로그램으로 말해질 수 있는 것이 협업 도구인 wiki나 사진 등을 공유하는 수단이 flicker 등이다. 한편 사용자가 소프트웨어사용자체에 포커스를 두는 온라인으로 서비스되는 소프트웨어(Software as a Service, SaaS) 등이 웹 기반 소프트웨어로 범주화될 수 있다.

이와 같이 웹의 진화에 따라 소프트웨어 자체의 경계도 변화 확장되고 있다. 앞으로 발전할 웹 기반 소프트웨어의 변화의 폭과 깊이를 이해하기 위해서 2장에서 웹의 플랫폼화(Web as a Platform)를 분석하고, 소프트웨어의 역할이 변화에 대응하는 소프트웨어의 전략적 키워드를 제안해보도록 하겠다.

2. 웹 플랫폼

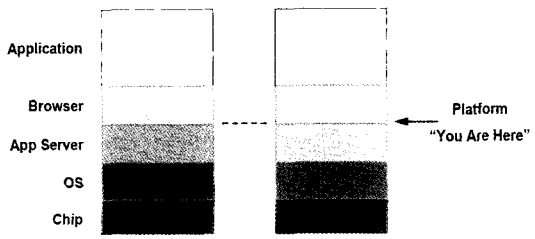
구글, 유튜브(YouTube), 마이스페이스(My Space), 플리커(Flicker) 등 다양한 웹 기반 서비스를 제공하는 기업이 등장하고 있다. 최근 이들 기업들은 이른바 Open API 정책을 통해 적극적

으로 자사의 생태계를 만들고 있고, 이들 기업들은 서비스 플랫폼 기업으로 불리기도 한다. 곧 서비스 기업의 핵심적 경쟁력이 플랫폼이란 소프트웨어 의존한다고 말할 수 있다.플랫폼으로 상징하는 소프트웨어와 서비스의 융합을 고찰하기 위해 플랫폼의 개념을 정의하고, MS같은 전통적인 플랫폼 기업과의 차별성을 분석해보도록 하겠다.

2.1 플랫폼의 정의

사실 플랫폼이란 말처럼 자주 다양하게 사용되는 개념도 없다. IT산업만 국한해서 보더라도 플랫폼은 마이크로프로세서나 OS에서부터 자바, 그리고 IBM의 웹스피어(Websphere) 같은 애플리케이션 서버까지 매우 다양하게 사용되고 있다. 그런데 최근 '웹 플랫폼'이나 '서비스 플랫폼' 유행하면서, 플랫폼의 의미가 더욱 더 혼란스러워 지고 있다.

플랫폼의 사전적 의미는 어떤 목적을 위해 높여진 단상이나 정치적 원칙이다. 이를 IT의 관점에서 해석해보면, 플랫폼은 수직적 스택(stack) 구조에서 수평적 레이어를 우선 의미한다. 또한 마치 대중을 향한 자신의 정치적 원칙을 표명하듯이, 그 수평적 레이어 위에서 솔루션이 개발되기 위한 인터페이스(interface)의 공개를 의미한다. 이를 하나로 정리해보면 플랫폼은 "솔루션이나 콘텐츠가 개발될 수 있도록 제공되는 인터페이스의 집합"이라고 정의될 수 있다.



(그림 1) 소프트웨어 스택과 플랫폼

이에 따르면 서로 다른 소프트웨어의 스택에서 각각의 레이어들은 서로 다른 레이어의 플랫폼이 된다. 예를 들면 CPU는 OS의, OS는 애플리케이션 서버의, 애플리케이션 서버는 웹 기반 애플리케이션의 플랫폼이다. 그런데 이 다양한 플랫폼을 열거하는 것이 우리의 목적은 아니다. 플랫폼을 하부기반(Infrastructure)과 구분해서 사용하는 것은 플랫폼이 IT산업의 다이나믹스를 나타내는 특별한 의미가 있기 때문이다. 없으면 안 되지만 가치생산에 있어서 핵심적 고리는 아닌 하부기반(Infrastructure)과 달리, 플랫폼은 인터페이스 정책이라는 수단을 가지고 생태계를 구축하는 산업적 영향력을 행사하고 있다.

2.2 PC시대 OS플랫폼

이제는 IT산업의 일상용어가 되어버린 플랫폼이란 용어가 대중적으로 알려진 것은 PC시대의 본격적 도래와 맞물려 있다. PC시대 플랫폼의 중심은 OS이다. MS는 OS란 플랫폼을 지배함으로써 IT산업에 대한 영향력을 확보할 수 있었다. MS의 플랫폼 지배는 자사의 OS를 둘러싼 하나의 생태계를 구축하였기 때문이다. 인터페이스의 집합으로써 플랫폼은 제품과 서비스간의 상호의존성을 전제로 한다. 그리고 그 상호의존성이 기업 내의 관계가 아니라 외부적인 관계일 때 비로소 의미가 있다. MS는 자사의 OS의 기반위에 응용프로그램이 만들어 질 수 있도록 API (Application Program Interface)를 공개함으로써, 소프트웨어산업의 수직적 분업을 이끄는 자사의 생태계를 구축하였다. 이점에서 API를 공개하지 않고 모든 것을 내부화한 애플과는 대조적인 모습이다.

API를 공개수준을 정할 수 있는 권한이 실질적 영향력을 가지기 위해서는, 규모의 경제에 기반 한 생산-소비-유통의 네트워크를 만들어야 한다. 우선 자사의 플랫폼인 OS자체가 시장에서 지배력이 있어야 하고, 공개된 API에 기반 해 응

용프로그램을 만드는 개발자와 파트너 기업들이 다수 존재하여야 한다. 이러한 관계의 네트워크를 통해 바로 제품이 아니라 스탠다드를 만들어 내기 때문이며, 역으로 스탠다드는 네트워크를 통해서만 창출될 수 있고 유지될 수 있기 때문이다. 이러한 선순환의 구조를 갖는 생태계를 통해 MS는 PC시대 진정으로 유일한 소프트웨어플랫폼기업이 될 수 있었다.

2.3 웹의 진화와 웹 플랫폼

그렇다면 웹이 플랫폼화 된다는 의미는 무엇인가? 위에서 살펴본 전통적인 플랫폼의 해석에 기초하면, 웹 플랫폼이란 '웹에 기반 한 솔루션이나 콘텐츠를 개발하기 위해 공개되는 인터페이스의 집합'이다. 또한 API를 공개하여 소비자, 개발자, 관련 기업들을 네트워크로 묶고자 하는 생태계가 출현한다는 의미가 더해질 수 있다. 웹이란 용어가 다소 부정확한 표현이지만, 실제 웹이 플랫폼화 되는 다양한 사례들이 등장하고 있다. 우선 최근 다양한 웹기반 서비스가 활성화되면서 웹 기반 애플리케이션의 기술적 토대가 되는 콘텐츠 신디케이션(Contents Syndication), 매쉬업(Mashup) 같은 다소 일반적 기술이나 Ajax, REST, RSS등 특정한 기술들이 등장하고 있다. 클라이언트 단 또는 서버 단에 위치한 패키지화된 특정 소프트웨어와는 다르지만, 이들 웹을 둘러싼 기술의 집합자체가 하나의 레이어와도 같은 의미를 가지게 되었다.

이처럼 웹이 하나의 하부기반(Infraxtructure)이 되면서 API 공개를 통해 자사가 중심이 된 생태계를 구축하려하는 포탈들이 등장하고 있다. 아마존, 이베이, 구글, 세일즈포스닷컴 등 수많은 미국의 다양한 포탈들은 자사의 데이터, 서비스, 솔루션을 공개하여 소비자가 이들을 조합하여 새로운 기능과 서비스들을 만들 수 있도록 하고 있다. 우리나라도 네이버, 다음, 씽크프리 등이 Open API정책을 추구하고 있다.

자사의 생태계를 구축하고자하는 이들 기업의 전략이 성립하기 위해선 우선 플랫폼을 제공해야 한다. 단순히 소비자가 원하는 서비스나 솔루션이 아니라 그 위에서 새로운 서비스와 솔루션을 추가할 수 있는 플랫폼을 제공해야 한다. 구글의 경우 가장 성공적인 서비스인 검색엔진 그 자체만으로는 플랫폼이 될 수 없지만, 다양한 매쉬업 서비스를 가능하기 하는 구글 맵(Maps)은 플랫폼으로 기능하고 있다.

플랫폼 전략이 성공하기 위해서는 소비자, 개발자, 관련 기업들을 참여시키는 다양한 관계의 네트워크를 구축하여야 한다. 이러한 네트워크를 통해 자사의 생태계를 확장할 때 API정책이 실질적 힘이 될 수 있다. 예를 들면 세일즈포스닷컴은 온라인 시장인 앱익스체인지(AppExchange)와 개발플랫폼인 애이팩스(Apex)를 통합한 플랫폼을 제공하여, 현재 200여개의 파트너사들과 1,5000여명의 개발자들이 참여하는 자신만의 생태계인 'The Business Web'으로 발전시키고자 한다. 또한 자사의 CRM과 구글의 애드센스의 연계하여 구글 애드센스를 통해 누가 무엇을 사는지에 대한 구매정보를 수집·가공하여 고객관리에 이용하도록 하였다.

2.4 OS 플랫폼과 웹 플랫폼의 차이

그렇다면 웹 2.0시대 웹 플랫폼은 PC시대 전통적인 플랫폼은 어떤 점에서 차이가 있을까? 플랫폼자체, 플랫폼위에서 만들어지는 솔루션·서비스, 생태계의 특성에서 다른 점이 있다. 우선 플랫폼자체를 보면, 앞에서도 언급했듯이 전통적 플랫폼은 제품으로 구매되는 하나의 소프트웨어를 의미한 반면에 웹 플랫폼은 인터넷을 통해 전달되는 다양한 소프트웨어 기술의 집합체이다. 또한 플랫폼위에서 만들어지는 것은 단순히 오피스, ERP 같은 전통적인 응용프로그램이 아니라 우리의 상상력을 뛰어넘는 다양한 서비스들이 제공되고 있다. 마지막으로 글로벌하

게 연결된 웹이란 네트워크의 특성 때문에 생태계의 범위나 동학이 다르다.

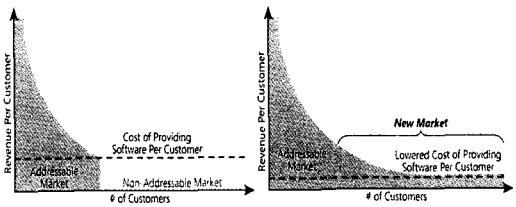
플랫폼이란 기본적 특징을 가지고 있지만 이러한 차이들 때문에 기존 소프트웨어 산업의 동학이 유지되면서도 또한 변하고 있다. 그동안 하나의 산업으로써 소프트웨어 산업의 발전을 경험하지 못한 우리나라에서 플랫폼 전략이나 생태계의 중요성이 기업가적 본능이나 기업전략의 상식으로 이해되고 있지 못하다. 따라서 새로운 패러다임에서 우리기업이 성공하기 위해서는 무엇보다도 플랫폼전략 자체의 중요성을 인식하고, 플랫폼 전략을 수립하기 위한 기본 방향을 공유해 보는 것이 필요하다. 3장의 소프트웨어산업의 키워드를 분석해 보면서 이를 더 자세히 살펴보도록 하자.

3. 소프트웨어 전략의 핵심요소

3.1 규모의 경제와 소프트웨어 개발

PC시대 소프트웨어를 제품화(Industrialization)한 패키지 소프트웨어산업에서 규모의 경제는 기본원칙이었다. 대량판매를 목적으로 하는 규모의 경제를 실현하기 위해 MS는 제품개발과 정보공개, 가격정책, 영업정책 등을 구사하였다. 이와 같은 규모의 경제가 개인소비자와 서비스의 시대인 웹 2.0시대에도 여전히 유효한 경제원칙일까?

구글의 검색광고에서 볼 수 있듯이 오늘날 검색포탈의 비즈니스 모델의 기반은 대규모의 트래픽이다. 또한 아래 그림에서 볼 수 있듯이 소프트웨어시장에서 롱테일 시장기회를 차지하기 위해서 가격경쟁력은 기본이다. 그렇지만 그 어떤 기업이 대량판매를 원치 않겠는가? 이러한 비즈니스 모델적인 요구가 과연 대규모생산에 따른 비용감소, 곧 규모의 경제를 반드시 필요로 하는 것일까? 다양하고 까다로운 개인소비자들의 취향을 만족시키기 위해서는 오히려 다품종 소량경제로의 진입해야하는 것은 아닌가?



(그림 2) 롱테일 시장기회와 비용감소

(자료: 마이크로소프트, 2006)

이러한 질문에 대해 입장이 정리되지 않고 혼란스러워 하는 것이 작금의 현실이다. 이 질문에 대답하기 위해선 소비자들의 구매행태와 생산비용에 대해서 다시 주목을 해보자. 오늘날 다양한 웹 2.0기업들은 서비스 이용은 하지만 꼭 구매하지는 않는 소비자를 유인해야 하는, 혹은 구매능력이 제한적인 다양한 형태의 롱테일 시장을 공략해야만 한다. 따라서 서비스 제공비용에 분명한 제한이 있을 수밖에 없다. 뿐만 아니라 대규모의 서비스 제공을 목표로 하기 때문에 규모에 따라 비용이 함께 증가해서는 수익이 나지 않는다. 가장 단순한 예로 구글은 대규모의 트래픽으로 인한 서버비용을 관리하기 위해 규모의 경제를 추구할 수밖에 없다. 곧 대규모의 서비스를 무료로 혹은 저가로 제공해야 하는 웹 2.0시대 규모의 경제는 여전히 기본적이고 유효한 덕목이다.

그렇지만 규모의 경제를 이루는 방법은 변화해야 한다. 지금은 서비스 시대이다. 따라서 똑같은 제품이 아니라 소비자에 따라 달라지는 서비스에서 규모의 경제를 추구해야 한다. 그러나 서비스에서 규모의 경제란 무엇일까? 대량생산을 위한 표준화와 하나의 고객을 위한 커스토마이제이션(customization)은 서로 개념적으로 모순적이지 않은가? 고객의 수요를 맞추는 방식이 변화해야 한다. 예를 들면 SaaS를 위한 소프트웨어개발방식은 패키지 소프트웨어를 개발하는 방식과 달라져야 한다. 패키지 소프트웨어의 개발방식이 조각 그림 맞추기(jigsaw puzzle)처럼

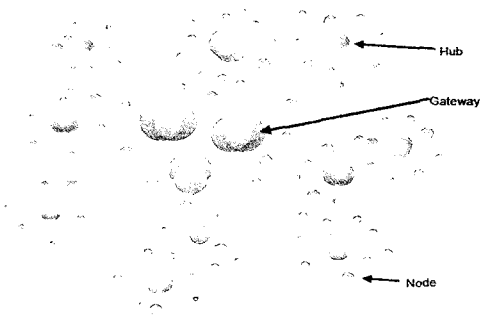
한 가지 그림만을 맞출 수 있는 모듈화가 아니라, SaaS를 위한 소프트웨어개발은 레고처럼 개인이 자유롭게 모양을 만드는 모듈화가 되어야 한다. 서비스시대에 맞는 규모의 경제를 달성하기 위해선 소프트웨어개발이 'customization'에서 'configuration'로 바뀌듯이 비즈니스 모델도 바뀌어가야 한다.

3.2 네트워크효과와 기업협력

메트카프의 법칙 (Metcalfe's Law)이 있다. 네트워크의 가치가 사용자의 수의 제곱에 비례한다는... 곧 두 명의 사용자는 한명의 사용자 보다 두 배가 아니라 네 배의 네트워크 효과를 만든다. 우리가 현재 목도하고 있는, 사용자가 만드는 모든 변화들의 효과는 바로 이 네트워크의 법칙으로 설명될 수 있다. 돌이켜 보면 PC시대 그렇게 강력했던 MS가 구축한 네트워크 효과도 연결되지 않았던 PC들의 숫자에 기초한 것이었다. 이제 인터넷은 전 세계의 PC를 연결시켰고, 사용자가 많으면 많을수록 더욱 더 강력해지는 네트워크 효과를 누리기 위한 전략수립의 중요성을 굳이 강조할 필요조차 없다.

네트워크 경제 하에서 비즈니스의 관계는 일면적이 아니라 다면적이 된다. 전통적인 비즈니스에서 생산에서 소비까지는 하나의 연속적인 흐름으로 이어졌다. 이에 비해 네트워크 경제에서는 그 흐름이 다면적으로 바뀌어가게 된다. 이는 기업이 네트워크의 부분적인 참여자로써 전문화됨에 따라 그 동안 내부화되었던 기능을 점차 다면적인 네트워크 관계로 이전하게 된다는 뜻이다. 생산과 유통의 일부만을 인터넷에 의존하는 전통적인 제조기업들 조차도 이미 새로운 비즈니스 패러다임에 직면하고 있다.

웹 2.0시대 네트워크의 의미를 분석하기 위해 우선 웹이라는 네트워크의 구조를 살펴볼 필요가 있다. 웹에서 네트워크는 구글, 야후, 네이버처럼 웹으로 들어오는 관문을 제공하는 게이트웨이(gateway), 이용자의 최종 목적지로써 다양한 컨텐츠의 공급자인 노드(node), 센터로써 독자적인 네트워크를 형성할 수 있는 플랫폼에 비유될 수 있는 허브(hub)로 구성된다.



(그림 3) 웹 경제의 네트워크 구조

네트효과 효과의 수혜자가 되기 위해서는 폐쇄적인 아닌 보다 많은 네트워크에 오픈되는 전략, 곧 경쟁전략만이 아니라 다양한 형태의 협력전략을 필요로 한다. 성공적인 노드가 되기 위해서는, 개인이나 중소기업과 같은 롱테일 시장기회를 잡기 위해 게이트웨이 혹은 허브에 연결되어야 한다. 독자적 생태계를 갖는 허브라는 궁극적 성공을 위해서는, 단순히 트래픽을 모으는 게이트웨이전략과는 차별된 오픈 API를 통한 플랫폼전략이 필요하다.

그런데 모든 기업들이 디지털 허브를 추구한다면 허브의 충돌은 없을까? 다시 메트카프의 법칙으로 돌아가자. 메트카프 법칙이 상정하고 있는 네트워크는 일종의 분산형 (peer to peer) 네트워크이다. 따라서 다수의 소규모 네트워크보다 하나의 대규모 네트워크가 더욱 강력한 네트효과 효과를 만든다. 이는 허브를 가진 각각의 네트워크가 마치 전화회사의 네트워크처럼 연결될 때 최대한의 가치를 발생한다는 의미이다. 이를 추구하기 위해서는 소프트웨어레벨에서 이 소프트웨어 사이의 장벽을 없애야 하고, 기업레벨에서 보면 허브의 폐쇄성이 아니라 유비쿼터

스하게 열린 네트워크 전략으로 가야 한다. 다시 강조한다면 네트워크의 수혜자가 되기 위해서는 소규모의 네트워크의 허브에 국한되지 않고 경쟁적인 허브들과 제휴(alliance)해가는 협력적 경쟁(co-opetition)이 기본 전략이 되어야 한다.

3.3 생태계 구축과 오픈 소스 커뮤니티

글로벌한 거대 네트워크의 허브가 되기 위한 플랫폼 경쟁은 수면아래서 너무도 격렬하다. 지금의 플랫폼 경쟁은 소프트웨어 플랫폼과 서비스 플랫폼의 경쟁, 서비스 플랫폼 사이의 경쟁으로 구분될 수 있다. 우선 마치 IBM에서 MS로의 이행과정처럼 새로운 서비스 플랫폼이 자유롭게 등장하기 위해서는, 기존 플랫폼의 중립화가 필요조건이다. 유일한 소프트웨어 플랫폼 기업인 MS의 OS 플랫폼은 필요하지만 지배적이지 않는, 단지 소프트웨어의 스택의 하부구조(infrastructure)로만 남아야 한다. 이를 위해 구글처럼 서비스 플랫폼기업은 플랫폼의 패러다임 전환이라는 이슈에서 자유로울 수 없다.

빌게이츠가 말한 'software plus service'라는 문구가 대변하듯이, MS는 오래된 비즈니스(혹은 플랫폼)와 새로운 비즈니스(혹은 플랫폼)를 동시에 추구하고 있다. 기업이윤의 대부분을 윈도우 OS판매와 오피스에 의존하는 상황에서, 너무나 자연스런 선택은 또한 전형적인 성공의 딜레마(inovator's dilemma)의 원인이 될 수도 있다. 예를 들면 MS가 Ajax의 기초가 되는 기술 대부분은 개발했고 처음으로 제품에 적용시켰지만, 이를 비즈니스로 성공시킨 것은 구글이다. 마치 그동안 다른 기업의 기술을 시장화한, 주로 두 번째 주자였던 MS를 보는 듯하다. 이미 중년이 넘어버린 MS가 참여·공유·개방이라는 웹 2.0의 젊은 비즈니스 논리에 적응하는 것은 쉬운 과제는 아마도 아닐 것이다.

그러나 새로운 시장기회를 찾는 후발자에게, 여전히 전 세계 96%의 데스크톱 OS에 기초 해 구축한 MS의 소프트웨어생태계는 위협이다. 윈도우 비스타와 WPF (Windows Presentation Foundation)의 연동에서 볼 수 있듯이, MS는 웹 플랫폼을 OS 플랫폼의 대체재가 아니라 보완재로 활용하고 있다. 또한 MS는 이 통합된 플랫폼으로 MS계열의 소프트웨어개발자를 끌어안으면서, 비 MS계열의 개발자에게 포용하는 방안을 모색하고 있다. 이렇게 새롭게 자사의 생태계를 업그레이드 하고자 하는 MS의 정책이 성공한다면, MS의 API 정책이 웹 2.0의 생태계를 만드는 주요 동인이 될 것이다.

따라서 MS 소프트웨어 플랫폼의 중립화그리고 소프트웨어개발자 커뮤니티는 새로운 생태계를 구축하기 위한 선결조건이다. 리눅스와 오픈소스 개발자커뮤니티는 이 조건을 충족시키기 위한 아마도 가장 현실적인 대안일 것이다. 오픈소스에 역사적인 연원을 둔 웹 2.0 시대에서, 오프소스의 창의적이고 유연한 활용은 역시 빠질 수 없는 기업전략임이 틀림없다.

4. 결 론

웹이 진화에 따른 소프트웨어의 변화를 관찰해보면, 소프트웨어산업의 구조적인 변화를 예고하는 징후들이 여기저기에 보인다. MS는 OS를 통해 독립적인 산업으로 패키지 소프트웨어 산업을 잉태시켰고, IT산업의 수직적 분업체계를 완성하였다. 지금 우리가 목격하고 있는 소프트웨어와 서비스의 융합은 웹 기반 소프트웨어라는 새로운 범주의 소프트웨어를 만들고 있고, 나아가 미래 IT산업의 범위와 분업구조까지도 변화될 것으로 보인다.

이러한 웹 기반 SW의 등장에 따라 변모되는 시장에 적응하기 위해 향후 소프트웨어기업은 다음의 세 가지 전략적 방향을 반드시 고려해야 한다. 첫째, 서비스 시대에 맞는 규모의 경제를 추구해야하는데, 레고처럼 개인이 자유롭게 모양을 만들 수 있는 모듈화를 추구하는 소프트웨

어개발방식이 그것이다. 둘째, 웹 2.0시대 더욱 더 강력해진 네트워크 효과를 최대한 이용해야 하는데, 다양한 기업들과 제휴하는 협력적 경쟁(co-petition)전략이 그것이다. 셋째, 자신의 플랫폼(혹은 생태계)를 구축하는 전략이 반드시 필요한데, 이를 위해 기존 소프트웨어플랫폼을 중립화시키고 소프트웨어개발자 커뮤니티를 유인할 수 있는 오픈소스 활용전략이 그것이다.

참고문헌

[1] 문장원, "리눅스 데스크톱: 레드오션 속의 블루오션", SW Insight 정책리포트, 2006년 8월호

[2] 백영란, 소프트웨어시장의 반독점 이슈, 한국소프트웨어진흥원 정책연구, 2004.

[3] 백영란, "웹 2.0의 유행을 넘어 SW산업의 키워드 찾아", SW Insight 정책리포트 2007 2월호

[4] 정제호, SaaS 시장분석과 정책방향, 한국소프트웨어진흥원 정책연구, 2006.

[5] Frederick Chong and Gianpaolo Carraro, "Architecture Strategies for Catching the Long Tail" (Microsoft, April 2006) http://msdn2.microsoft.com/en-us/library/aa479069.aspx

[6] Jack Greenfield, The Case for Software Factories, Microsoft Architect Journal (July, 2004) http://msdn2.microsoft.com/en-us/library/aa480032.aspx

[7] Gartner, "Microsoft Courts Broader Community of Ajax Developers With Changes to Atlas", 2006.

[8] Gartner, Web 2.0 Poses a Threat and an Opportunity to Microsoft, 2006.

저자약력

백 영 란

1987년 8월 서울대학교 인문대학 국사학과 학사 졸업
1992년 8월 서울대학교 인문대학 국사학과 석사 졸업
2002년 3월 UCLA 경제학과 박사
2001년~2002년 UC Riverside에서 Lecturer
2003년 대통령인수위원회 경제 2분과 자문위원
2004년~2005년 공정거래위원회 IT분과 자문위원
2006년~현재 외교통상부 한미FTA 전문가 자문위원
2007년~현재 공정거래위원회 MS이행감시 자문위원
2002년 8월~현재 한국소프트웨어진흥원, SW정책연구센터
이 메 일 : ynbaek@software.or.kr