

사회적 소프트웨어(Social Software)와 지식관리 프로세스

Social Software and Knowledge Management Process

김학래^a, 최재화^b, 김흥기^c

^a 단국대학교 일반대학원 경영 정보학과
서울 용산구 한남로 147, 140-714
Tel: +82-41-550-3368, E-mail: hlkim@dku.edu

^b 단국대학교 경상대학
충남 천안시 안서동 산 27, 330-714
Tel: +82-41-550-3368, E-mail: jchoi@dankook.ac.kr

^c 서울대학교 치과대학 의료지식공학 연구실
서울 중구 연건동 147, 110-460
Tel: +82-2-740-8793, E-mail: hgkim@snu.ac.kr

초록

정보근로자는 지식 중심의 사회에서 지식을 창출하고 소비하는 핵심적인 역할을 하고 있다. 따라서 지식관리 과정은 정보근로자의 업무패턴을 반영해야 한다. 또한 지식관리도구는 쉽고 효과적으로 지식을 생성, 통합, 전파할 수 있도록 지원해야 한다.

그러나 대부분의 지식관리 시스템은 정보근로자를 시스템의 기능으로 다루고 있기 때문에 정보근로자의 업무패턴과 요구사항을 반영하는데 어려움이 있다.

본 연구에서는 정보근로자의 지식관리를 위해 요구되는 협업, 정보교환, 정보접근의 중요성을 설명하며, 사회적 소프트웨어 기반의 지식관리 프로세스와 지식 변환 과정을 살펴본다.

키워드:

사회적 소프트웨어, 웹2.0, 협업, 정보교환, 지식관리, 블로그, 위키, 포크소노미

서론

정보근로자(Information worker)는 지식 중심의 사회에서 지식을 창출하고 소비하는 핵심적인 역할을 하고 있다. 일반적으로 정보근로자는 지식근로자(knowledge worker)보다 광범위한 개념으로 조직 내부 뿐만 아니라 외부의 파트너도

포함하는 의미로 사용되고 있다. 한편 정보근로자의 업무는 점점 전문화되고 세분화되고 있다. 정보근로자는 전자우편, 문서작성, 전화통화, 미팅과 같은 활동에 업무 시간의 대부분을 사용하고 있다. 이러한 활동은 정보근로자의 업무 형태가 협업(collaboration)과 정보교환(communication)을 중심으로 하고 있음을 의미한다.

협업과 정보교환은 조직의 생산성 향상을 위한 핵심 요소로 언급되고 있다. 과거 협업의 개념은 개인과 팀, 혹은 조직내로 범위가 한정되었다. 그러나 정보기술의 발전으로 협업의 대상은 외부의 파트너 뿐만 아니라 고객으로 까지 확대되고 있다. 따라서 정보근로자는 과거보다 다양해진 구성원들과 협업해야 되고 이들과 정보교환을 통해 지식을 관리해야 한다.

일반적으로 지식관리 프로세스는 지식을 창출하고 통합하고 전파하는 순환적인 과정을 갖고 있다. 조직내에서 정보근로자는 지식관리를 위해 KMS와 같은 특정한 시스템을 사용할 수 있지만, 지식관리를 원활히 하는데 제약이 많다. 마이크로소프트 리서치(2002)에 의하면, 정보근로자의 91%가 주기적으로 메모를 하고, 이중 26%는 노트에 기록한 메모를 전자우편으로 전환한다고 한다. 즉, 대부분의 정보근로자는 자신의 아이디어나 중요한 일정을 종이와 펜을 이용하여 관리하고 있음을 알 수 있다. 그러나 종이에 기록된 아이디어는 컴퓨터를 이용해 검색하는 것과 같은 성능을 제공하지 않는다.

정보근로자는 지식관리를 위해 데스크탑 및 웹

환경 기반의 다양한 도구를 활용할 수 있다. 그러나 지금까지 개발된 대부분의 도구들은 정보근로자를 시스템의 단일 기능으로 다루고 있기 때문에 정보근로자의 업무패턴과 요구사항을 반영하는데 어려움이 있다.

지식관리 프로세스에서 협업 환경과 정보 교환을 효과적으로 지원하기 위해 정보근로자가 시스템의 핵심적인 역할을 하는 것이 중요하다. 비록 자동화된 시스템이 지식의 일부를 만들고 통합할 수 있지만 조직의 지식은 정보근로자의 경험과 이해로 생산된다. 인센티브와 같은 정책적 지원도 정보근로자를 지식관리 프로세스에 끌어들이는데 한계가 있다. 따라서 정보근로자가 스스로 자신의 지식을 관리하고 협업할 수 있는 환경이 무엇인지 분석하고, 이를 제공해 주는 것이 문제 해결의 핵심이라고 할 수 있다.

본 연구에서는 정보근로자의 협업, 정보교환, 정보 접근을 위한 사회적 소프트웨어의 중요성을 설명하고, 사회적 소프트웨어가 지식관리 프로세스에 어떤 영향을 줄 수 있는지 살펴본다.

사회적 소프트웨어의 개요

정의 및 진화 과정

사회적 소프트웨어(Social Software)는 개인의 존재(identity)를 바탕으로 온라인에서 사회적 관계와 상호작용을 통해 인맥을 형성할 수 있도록 지원하는 시스템 또는 서비스이다. 시스템 측면에서 본 사회적 소프트웨어는 사람과 사람, 사람과 정보를 연결하고 이들간의 공동작업을 지원하고 정보 교환을 가능하게 하는 도구라 볼 수 있다. 사회적 소프트웨어는 협업과 정보교환의 도구로 블로그(1대 다), 전자 우편(1대 1), 메신저(1대 1 또는 다), 위키(다 대 다) 등을 포함하며, 이러한 도구를 연결하기 위해 사용되는 AJAX, RSS/ATOM, 포크소노미(folksonomy) 등도 그 범주에 포함시킬 수 있다. 궁극적으로 정보교환의 도구로서 사회적 소프트웨어는 다대다(many-to-many)의 관계로 확장될 수 있다.

사회적 소프트웨어에 대한 개념은 바네바 부시(Vannevar Bush)가 “메멕스(memex)”를 언급한 1945년에 시작되어 60년대의 확장(Augmentation), 70년대의 사무 자동화(office automation), EIES(Electronic Information Exchange System), 80년대의 CSCW(Computer-Supported Collaborative Work), 90년대의 그룹웨어(Groupware)로 발전되어 왔다. 부시가 ‘As We May Think[12]’에서 언급한 메멕스는 사람의 기억력과 사고능력과 비슷한 기능을 수행할 수 있는 가상의 기계이다. 부시는 정보를 확장할 수 있는 방법보다 저장되어 있는 자원을 찾는 일이 더 중요한 것임을 강조하고 있다. Memex는 컴퓨터

과학에 있어 선구적인 발상을 제공하였는데, 웹의 창시자인 팀버너스리는 웹이 메멕스를 구현한 일종의 시스템이라고 소개하고 있다[17].

부시의 아이디어는 하이퍼텍스트(hypertext)라는 용어를 만들어낸 엔겔바트(Douglas Englebart)에 의해 실현화되었다. 엔겔바트[13]는 지적능력의 확장에 대한 개념을 정의하며 사람의 문제해결은 소프트웨어에 의해 발전될 것으로 예측하고 있다.

최초의 협업 도구인 EIES를 만든 머레이(Murray Turoff)는 컴퓨터 매개 정보교환이 양방향적인 상호관계로 정의하며, 컴퓨터를 바탕으로 그룹간의 정보 교환이 협업을 위한 시간적/공간적 제약을 해결할 수 있다고 주장하고 있다. 피터와 트루디(Peter & Trudy Johnson-Lenz)는 최초로 그룹웨어(groupware)라는 개념을 사용했으며, 이를 ‘그룹 프로세스와 그룹 구성원을 지원하기 위한 소프트웨어’로 정의하였다[16].

사회적 소프트웨어라는 용어가 처음 등장한 것은 1990년으로 ‘open hypertext’와 ‘social-hyper computing’과 관련된 개념으로 소개되었다. 이후 1992년에 넬슨(Nelson)의 Xanadu와 필 살린(Phil Salin)의 AMIX를 사회적 소프트웨어로 정의하였다.

사회적 소프트웨어가 본격적으로 알려진 것은 2000년 후반부터이다. 1995년에 소개된 위키(wiki)가 커다란 인기를 끌지 못했지만, 2000년 이후 정보기술과 웹 기반 환경의 급속한 발전은 컴퓨터를 이용한 협업 작업을 가능하게 했다. 아울러 최근 급격히 증가하고 있는 위키와 블로그의 이용이 사회적 소프트웨어의 가능성을 구체화시키고 있다.

사회적 소프트웨어의 특징

앞서 언급한 바와 같이 사회적 소프트웨어는 아주 오랫동안 언급되고 진화된 시스템이라 할 수 있다. 웹이 하나의 컴퓨터이고 거대한 메모리라고 할 때 사회적 소프트웨어는 이를 기반으로 사람들의 집단 지성(collective intelligence)을 만들어 낼 수 있는 도구라 말할 수 있다.

사회적 소프트웨어는 분산되어(decentralized) 있는 웹 자원을 사람을 중심으로 연결시킬 수 있으며(social networking), 연결된 네트워크 사이에 영향을 주고(feedback), 스스로 조직을 구성하고 확장할 수 있는(self-organized) 특성을 기반으로 상향식(bottom-up) 협업 프로세스를 유도할 수 있다.

사회적 소프트웨어는 구성원의 자율적인 참여를 기본으로 한다. 예를 들어, 온라인 백과사전인 위키피디아¹는 누구나 참여해 콘텐츠를 작성하거나 수정할 수 있다. 딜리셔스²나 플리커³는 구성원이 참여하여 공개된 모든 자원에 새로운 꼬리표(tags)를 만들고 이를 통해 포크소노미(folksonomy)를

¹ <http://en.wikipedia.org>

² <http://del.icio.us>

³ <http://www.flickr.com>

진화시킨다. XFN(XHTML Friends Network)이나 FOAF(Friend of a friend)는 하이퍼링크를 이용해 사람들의 관계를 표현할 수 있는 방법을 제공한다.

사회적 소프트웨어와 웹2.0

블로그, 위키와 같은 웹 기반 사회적 소프트웨어는 최근 관심이 집중되고 있는 웹2.0(Web2.0)과 밀접한 관계를 갖고 있다. 위키피디아(Wikipedia)는 웹2.0을 “사람들이 온라인에서 정보를 공동으로 만들고(collaborate) 공유할 수(share) 있게 지원하는 월드와이드웹의 제 2세대 서비스(second generation of services)”라고 정의하고 있다. 팀오라일리(Tim O'Reilly)는 “차세대 소프트웨어에 대한 디자인 패턴과 비즈니스 모델이라고 정의하며, 1) 플랫폼으로서의 웹, 2) 지성/지혜를 모으는 작업, 3) 차세대 인텔 인사이드는 바로 데이터, 4) 궁극적인 소프트웨어, 5) 경량 프로그래밍 모델, 6) PC를 벗어난 장비에서 작동하는 소프트웨어, 7) 풍부한 사용자 경험 등의 7가지 핵심 요소를 제시했다[18].

즉, 웹2.0 기반의 서비스는 좀 더 가볍고 풍부한 사용자 인터페이스를 이용하여 사용자가 쉽게 데이터를 만들고 공유할 수 있는 환경을 지향한다.

사회적 소프트웨어는 웹2.0의 핵심 기술로 언급되기도 하고, 상호보완적 관계로 보는 관점도 있다.

예를 들어 블로그는 신디케이션(syndication) 표준을 이용해 콘텐츠를 배포할 수 있다⁴. 신디케이션은 사이트 콘텐츠의 일부 또는 전체를 다른 서비스에서 이용할 수 있도록 해주는 것을 말한다. 블로그는 RSS/ATOM 표준을 사용하고 있으며, 이것은 웹2.0의 중요한 데이터 교환/전달 기술중에 하나이다. 웹2.0은 학문 영역이 아닌 웹에서 시작되어 진화하고 있으며, 현재까지 명확한 정의가 만들어진 것은 아니다. 따라서 웹2.0은 다양한 기술과 개념을 모두 포함하고 있는 개념이라 할 수 있다. 이런 측면에서 웹 2.0은 사회적 소프트웨어의 기반 구조라 정의할 수 있다.

사회적 소프트웨어와 지식관리

지식 관리 프로세스는 지식을 생성(creation), 통합(integration), 전파(dissemination)하는 순환적 구조를 갖고 있다. 지식은 정지된 상태로 머물지 않고 메타지식(meta knowledge)과의 관계를 통해 계속 진화하게 된다[1]. 그러나 전통적인 지식관리시스템은 개인과 조직이 갖고 있는 지식을 저장하고 관리하는데 목적이 있었기 때문에 지식 저장소를 통해 생성된 지식도 사용자의 요구를 만족시키기 어렵다. 전통적인 접근은 지식을 생성하고 관리하기 위한 효과적인 방법으로 사람 중심이 아닌 시스템의 성능 향상에 목적을 두는

하향식(top-down) 접근 방법이었다. 따라서 이런 형태의 지식은 인간의 심성모형(mental model)을 반영하지 못하며, 재사용되거나 다른 형태의 지식으로 전파되기 어렵다. 그렇다면 사회적 소프트웨어는 어떻게 지식관리를 인간중심으로 전환할 수 있을까?

인간 중심의 지식관리는 시스템이 아닌 사람의 관계를 기본으로 접근하는 상향식(bottom-up) 접근방법이다. 지식은 개방된(open) 시스템 환경에서 사람들의 협력적인 관계로 부터 생성된다. 지식 저장소는 지식 생성과 전파의 지속적인 프로세스를 지원한다. 또한 구성원의 조직 활동으로 생기는 부산물(by-product)도 포함할 수 있다. 따라서 지식관리는 사람들의 관계를 효과적으로 연결하기 위한 방법으로 재정의 할 수 있다.

전통적인 지식관리시스템은 처음부터 엄격한 규칙과 기준에 의해 만들어진다. 조직 구성원은 자신의 소속 또는 업무에 따라 다른 권한을 부여 받고, 규정된 공간에서만 활동할 수 있다. 이러한 시스템은 사람을 하나의 기능으로 취급하기 때문에 개인의 다양한 요구 사항을 반영하기 어렵다.

반면 사회적 소프트웨어 기반의 지식관리는 개인의 욕구와 관심을 바탕으로 지식 활동을 지원한다. 개인은 자신의 아이덴티티(또는 고유한 권리)를 갖고 사회적 네트워크를 만들며, 피드백과 다른 사람의 참여를 통해 네트워크를 확장시켜 나간다. 이러한 사회적 과정이 지식관리 프로세스로 수렴된다. 인간 중심의 지식관리는 다양한 관계로 부터 지식이 생성, 통합, 전파되기 때문에 시스템의 기능도 달라지게 된다. 표는 사회적 소프트웨어 기반의 지식 관리의 특징을 전통적인 방법과 비교하고 있다. 앞서 언급한 것과 같이 지식관리 기법의 가장 큰 차이점은 지식관리 주체가 누구인지에 있다. 지식의 생성과 전파는 소수 전문가에서 조직 구성원 전체로 확대되고 이에 따라 업무나 학습의 방식도 바뀌게 된다 [표 1 참고].

구분	전통적 지식 관리	사회적 소프트웨어 기반 지식관리
지식 생성	전문가, 전문 활동	누구나, 협업 활동
지식 통합	Design time	Use time
지식 전파	강의, 학습	온디맨드(On-demand) 업무와 관련된 학습과 작업의 동시병행 및 통합개인화(personalization)
학습 패러다임	지식전이(knowledge transfer)	지식 구성(Knowledge construction)
업무(tasks)	시스템 기반(System)	사용자 및 업무 기반 (User/task)

⁴ <http://www.w3c.or.kr/~hollobit/data/paper/TTA-RSS2.htm>

	driven(canonical)	driven(situated))
사회적 구조	계층화(hierarchical) 하향식(top-down)	실무공동체(Communities of practice) 상향식(bottom up) peer-to-peer
업무 패턴	표준화	진화형
정보 환경	폐쇄형(Closed), 정적(static)	개방형(Open), 동적(dynamic)
정보 접근	푸시(Push)	출판 및 구독(Publish and subscribe)

표 1 전통적 지식관리와 사회적 소프트웨어 기반 지식관리의 비교

사회적 소프트웨어 기반의 지식관리는 종전의 지식관리 프로세스를 보완할 수 있는 많은 변화를 요구한다. 지식관리 프로세스의 각 단계에 사람이 참여한다는 것은 이상적일 수 있다. 예를 들어, 지식의 생성은 개인의 선형적 지식에 의존하지 않고 협력적인 관계에서 얻을 수 있는 다양한 정보를 통해 만들어질 수 있다. 이 경우, 시스템은 개인이 협력적인 관계를 유지할 수 있도록 지원해야 한다.

사회적 소프트웨어와 지식관리 프로세스

지금까지 지식관리시스템은 지식 활용의 극대화를 위해 전사 차원의 통합 저장소를 만들고 포털을 통해 공유하려는 시도를 해 왔다. 뿐만 아니라 시스템에 지식을 넣기 위한 문화를 만들기 위해 마일리지, 포상과 같은 전략적 활동도 병행되어 왔다. 그러나 시스템의 도입과 다양한 전략적 활동도 정보근로자의 요구를 반영하는데 한계가 있다. 정보근로자의 업무패턴은 현존하는 시스템과 많은 차이를 보이고 있다. 지식관리의 가장 큰 문제점은 지식을 통합하고 전파하는데 있지 않다. 대부분의 정보는 정보근로자의 개인 컴퓨터나 머리 속에 존재한다. 정보근로자는 현재 시스템 환경에서 정보를 지식화시키기 위해 업무활동과 별도로 시간과 노력을 투자해야 한다. 결국 정보근로자는 다양한 마일리지와 포상에도 불구하고 지식화 활동을 피하게 된다. 이 문제는 위에서 지적했듯이 정보근로자가 시스템의 일부 혹은 기능이기 때문에 발생한다.

사회적 소프트웨어는 지식관리 프로세스의 각 단계에서 지식화를 도울 수 있다. 지식이 생성되는 과정은 협업(collaboration), 정보교환(communication), 정보 접근(information access)이라는 사회화 과정을 포함하고 있다. 정보근로자는 전자우편, 회의, 정보 검색 및 활용에 업무 시간의 대부분을 사용한다. 예를 들어, 정보근로자가 업무를 시작하기 위해 컴퓨터를 켜고 있을 때 가장 먼저 전자우편 관련 프로그램을 실행한다고 한다[19, 20]. 정보근로자는 전자우편을 확인한 후 1) 협업할 업무(collaboration), 2) 다른 사람으로부터 확인할 정보(communication), 3)

추가적으로 요구되는 정보(information access)가 있는지 점검하며 업무를 진행한다. 즉, 정보근로자의 업무패턴은 개인을 중심으로 협업, 정보교환, 정보 접근과 밀접한 관련이 있다. 그림은 정보근로자의 지식관리 프로세스가 어떻게 사회적 소프트웨어와 연결될 수 있는지 보여주며, 협업-정보교환-정보접근이 지식관리 프로세스의 핵심요소임을 설명하고 있다.

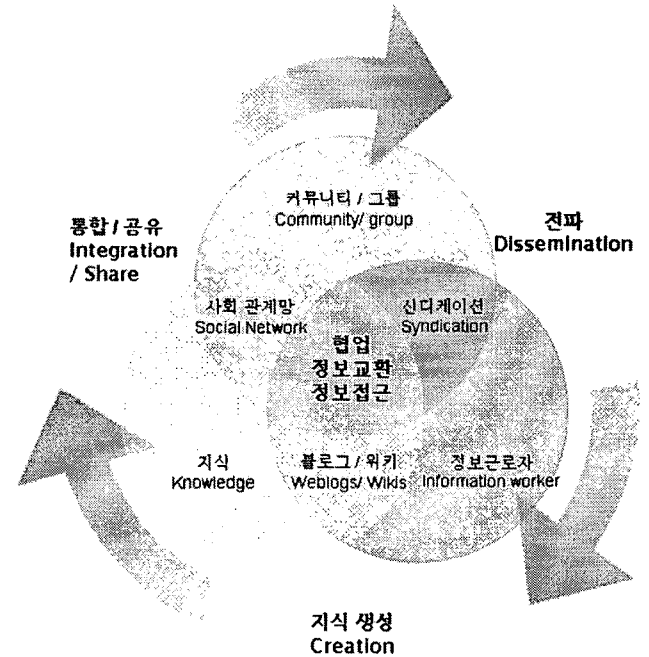


그림1 사회적 소프트웨어와 지식관리 프로세스

지식근로자가 생성한 지식이 커뮤니티를 통해 확산되는 과정은 블로그, 위키와 같은 사회적 소프트웨어를 기반으로 한다. 이 때 지식은 개인의 고립된 환경이 아닌 네트워크로 연결된 사람 혹은 채널의 피드백을 통해 만들어진다. 커뮤니티에 통합된 지식은 다시 정보근로자의 수요에 의해 전파될 수 있다. 정보근로자의 수요에 의한 전파는 정보 소비의 주체가 사람에게 있다는 것을 의미한다. 정보근로자는 정보를 얻기 위해 커뮤니티에 가입하기 위해 아이덴티티(예: 주민번호, 아이디, 비밀번호 등)를 노출할 필요없이 원하는 정보를 신디케이션을 통해 얻을 수 있다.

즉 지식관리프로세스는 전통적인 프로세스의 순환적 구조로 동일하지만, 사회적 소프트웨어는 인간을 중심으로 지식화 과정을 지원하게 된다.

사회적 소프트웨어의 지식화 프로세스

정보근로자는 시간순이 아닌 협업-정보교환-정보접근의 유기적 순환작업으로 지식화를 진행할 수 있다. 사회적 소프트웨어의 지식화 과정은 포크소노미(folksonomy), 포드캐스팅(podcasting), 신디케이션(syndication),

비동기 자바스크립트 XML(AJAX: Asynchronous JavaScript XML) 등과 같은 웹 기술을 기반으로 하고 있다. 즉, 이러한 웹 기술은 정보근로자의 지식화 과정을 실현하는 역할을 한다.

협업

정보근로자는 개인의 지식을 위키나 블로그를 이용해 명시적으로(explicit) 표현할 수 있으며, 다양한 기술을 바탕으로 다른 사람과 지식을 공유할 수 있다. 예를 들어, 포크소노미는 개인이 만든 꼬리표(Tags)를 집단의 그것과 비교하여 가중치를 부여해 집단 지성(community intelligence)으로 표현한 것이다. 꼬리표는 개인의 콘텐츠에 저장되는데 그치지 않고 정보를 재창조, 생산, 저장, 공유할 수 있는 기반이 된다. 플리커⁵나 딜리셔스⁶는 꼬리표를 이용해 정보를 표현하고 공유할 수 있는 대표적 사례이다.

대부분의 정보근로자는 분류 시스템을 사용해 정보를 공유하거나 접근하는데 어려움을 호소한다. 분류 시스템은 계층적인 구조를 갖고 있어 검색을 쉽게 할 수 있지만, 정보근로자의 심성모형과 다를 경우 이를 활용하는 것은 매우 어려워진다.

반면 꼬리표는 특정 콘텐츠에 대한 이해나 관점을 즉, 개인의 심성 모형을 표현할 수 있다. 꼬리표는 맥락에 따라 다양하게 표현될 수 있지만, 커뮤니티 구성원은 특정 꼬리표에 대한 일반적인 동의를 통해 확장할 수 있다. 결국 꼬리표는 구성원의 평판에 따라 진화하게 되며 이를 통해 맥락에 적합한 꼬리표의 집합인 포크소노미가 만들어진다. 정보근로자는 자신이 만든 꼬리표와 구성원의 참여로 만들어진 포크소노미를 비교하며 자신의 심성모형을 발전시킬 수 있다.

정보교환

대부분의 정보근로자는 정보교환 수단으로 전자우편을 사용하고 있다. 전자우편은 1) 다른 사람과 정보를 교환, 2) 정보를 구독할 수 있는 도구이다. 그러나 전자우편은 두 가지 핵심 기능에 대한 문제점을 갖고 있다. 첫째, 정보 교환 과정은 스팸 메일에 노출되어 있다. 전자우편 서비스 제공자가 스팸을 차단하기 위한 도구를 제공하고, 법적으로 스팸 메일에 대한 처벌 규정을 만들었지만 이를 개선하는 것은 쉽지 않다. 둘째, 소식지(news letter)를 구독하기 위해 개인 정보를 반드시 제공해야 하지만 구독을 해제하는 것이 쉽지 않다. 더구나 제공된 정보가 노출되기 쉽다.

사회적 소프트웨어는 정보교환의 기본 수단으로 RSS/ATOM을 사용한다. 전자우편이 정보제공자 중심의 서비스라면 RSS/ATOM 기반의 정보교환 방식은 정보 수요자 중심의 서비스이다. 즉, 정보제공자가 콘텐츠를 RSS/ATOM 표준 형식으로

제공할 경우, 정보 수요자는 RSS/ATOM 구독기를 이용해 해당 콘텐츠⁷를 읽어올 수 있다. RSS/ATOM 기반의 정보 교환은 1) 스팸 메일을 받지 않아도 되고, 2) 개인 정보가 노출되지 않으며, 3) 구독 혹은 해지가 쉽다는 장점을 갖고 있다.

사회적 소프트웨어 환경에서 정보근로자는 정보제공자와 정보수요자의 역할을 동시에 갖고 있다. 정보근로자가 생성한 지식은 정보 제공의 역할을, 다른 지식을 읽어오는 것은 수요의 역할과 일치한다. RSS/ATOM 기반의 서비스는 정보 제공과 수요의 역할을 동시에 지원할 수 있다. 예를 들어, 정보제공자가 블로그나 위키로 생성한 콘텐츠는 RSS/ATOM으로 공개될 수 있다. 정보수요자는 RSS/ATOM 구독기를 이용해 다른 콘텐츠를 자동으로 읽어올 수 있고 관리할 수 있다. 즉, 정보근로자는 자신이 원하는 정보만 얻을 수 있으며 자신의 정보를 구독하는 사람들의 특징을 분석해 전문화된 정보를 제공할 수 있다.

정보접근

사회적 소프트웨어는 정보를 하나의 저장소로 통합하려는 시도와 달리 분산화된 서비스를 기반으로 하고 있다. 정보근로자는 업무패턴과 접근하려는 정보의 특성에 따라 사람과 사람의 연결을 통해 혹은 사람과 정보의 연결을 통해 얻을 수 있다. 매쉬업(mashup)은 이러한 특징을 반영한 현상이다. 매쉬업은 음악 분야에서 주로 사용되어 왔는데, 2가지 곡을 조합하여 하나의 곡을 만들어 내는 것을 의미한다. 웹2.0 기반의 매쉬업은 복수의 서비스 혹은 API를 조합하여 새로운 서비스를 만들어내는 것을 의미한다. 하우스맵스⁸는 구글맵(googleMap⁹)을 이용한 매쉬업 서비스의 시초이며 매쉬업의 새로운 가능성을 보여주고 있다.

일반적으로 매쉬업 서비스는 간단한 API(Application Programming Interface)를 기본으로 하고 있다. 구글,야후와 같은 서비스 제공자는 공개(public) API를 무료로 제공하고, 서비스 수요자는 API를 이용하거나 개발된 서비스에 자신이 원하는 기능을 추가하여 서비스를 개발할 수 있다. 이러한 과정은 객체지향 프로그래밍 언어를 이용할 수 있지만, 대부분 AJAX로 불리는 자바스크립트 언어를 이용해 개발할 수 있다. 자바스크립트가 클라이언트를 위해 사용되었을 경우 대부분은 오픈소스(open source)의 성격을 갖는다. 이것은 공개 API와 더불어 매쉬업을 확장시킬 수 있는 주요한 특징이다.

정보근로자는 매쉬업 서비스를 이용해서 지식을

⁵ <http://www.flickr.com>

⁶ <http://del.icio.us>

⁷ 피드(feed): 블로그의 내용을 담고 있는 문서로 일반적으로 웹 피드 혹은 RSS/ATOM Feed를 줄여서 말함

⁸ <http://housingmaps.com>

⁹ <http://maps.google.com>

생성할 수 있고, 공개할 수 있다. 이 과정은 공개 API와 AJAX 기술을 이용해 비교적 간단히 구현할 수 있다.

사회적 소프트웨어 기술 기반의 지식 변환

앞서 언급한 바와 같이 사회적 소프트웨어 기반의 지식관리 프로세스는 협업-정보교환-정보 접근이라는 순환적인 지식화 과정을 통해 이루어진다. 유사한 측면으로, 그렇다면 정보근로자가 사회적 소프트웨어를 이용한 지식을 변환하는 과정은 어떻게 이루어지는가? 지식 변환 과정에 사회적 소프트웨어는 어떤 역할을 하게 될 것인가?

지식의 변환 과정을 언급하기 앞서 지식의 차원에 대한 먼저 살펴본다. 지식의 차원에 대한 연구는 지금까지 많은 연구자에 의해 수행되어 왔다. 먼저 Polanyi는 지식을 형식지(explicit knowledge)와 암묵지(tacit knowledge)로 구분한다. 형식지는 코드로 만들 수 있고 언어에 의해 공유할 수 있는 지식을 말하며, 암묵지는 의사소통이 어렵고 행위에 의해 표현할 수 있는 지식을 말한다. Anderson은 지식의 차원을 선언지(declarative knowledge)과 절차지(procedural knowledge)로 구분하고 있다. 선언지는 어떤 사실과 명제를 알고 있는지(knowing what)에 관련된 지식이고, 절차지는 어떤 일을 하는 방법을 알고 있는지(knowing how)에 관련된 지식이다[1, 2]. 일반적으로 절차지는 선언지에 비해 암묵적이고 더 느리게 잊게 되는 특성이 있다. Nonaka는 지식 창출 과정은 암묵지에서 암묵지, 암묵지에서 형식지, 형식지에서 형식지, 형식지에서 암묵지 등 4단계로 나선형 프로세스(spiral process)로 진행된다고 설명하고 있다.

사회적 소프트웨어의 기술은 Nonaka의 지식변환 과정의 각 단계에서 지식을 생성하고 변환하는 주체가 될 수 있다. 사람이 사용하는 대부분의 지식은 암묵지로 구성되어 있기 때문에 형식지로 조직하는 것이 매우 어렵다. 그러나 사회적 소프트웨어를 통해 정보근로자의 암묵지는 형식 지로 쉽게 변환될 수 있다.

그림 2는 사회적 소프트웨어가 지식변환 과정에 어떤 역할을 하고 있는 보여주고 있다. 지식의 변환과정에 있어 사회적 소프트웨어는 정보근로자의 업무패턴에 적합한 도구를 지원하며, 각 단계를 지식화 과정을 통해 지식의 변환을 도울 수 있다.

사회화(Socialization)

정보근로자는 개인이 갖고 있는 경험과 지식을 블로그, 위키를 통해 새로운 형태의 암묵지로 만들 수 있다. 블로그, 위키는 개인이 갖고 있는 암묵지의 구조를 변형하는 과정에 사용될 수 있고 이를 통해 콘텐츠를 만들 수 있다. 콘텐츠에 추가되는 꼬리표는 정보근로자가 갖고 있는 인지구조를 바탕으로 만들어진다. 이 때 지식은 정보근로자가 갖고 있는 암묵지를 새로운 형태의 암묵지로 만드는 과정이며, 형식지와 상호작용은 존재하지 않는다. 예를 들어, 정보사용자가 데스크탑에서 위키와 블로그를 사용할 수 있다. 데스크탑에서 사용되는 사회적 소프트웨어는 정보근로자의 통제하에 관리될 수 있으며, 원하는 지식만 공유할 수 있다. 생성되는 콘텐츠는 개인이 갖고 있는 경험과 아이디어를 정리하고 인지구조를 개선시키는 과정으로 볼 수 있다. 따라서 정보근로자가 각각의 콘텐츠에 꼬리표를 만들어도 이것이 사회적으로 공인되는 지식과 일치하지 않는다. 즉 이 때 만들어진 지식은 주관적인 지식에 한정된다.

표출화(Externalization)

암묵지가 조직이나 공동체에 공개되면 형식지로 변환한다. 이 과정은 정보근로자가 갖고 있는 암묵지를 공개하고 구성원과 상호작용을 위해 지식을 재구성하거나 새로운 형태로 지식구조를 변형시키는 과정이라 할 수 있다.

정보근로자가 사회적 소프트웨어를 통해 만든 지식은 신디케이션을 통해 형식지로 변환된다. RSS/ATOM은 형식지 형태로 전환된 지식을 담고 있으며 RSS/ATOM 구독기를 통해 공동체에 전파된다. 공동체의 구성원들은 특정 정보근로자의 콘텐츠와 꼬리표를 구독하고 재활용할 수 있다.

이 과정은 기존의 지식을 재구성할 뿐 아니라 피드백을 통해 지식을 활용할 수 있는 기초가 된다.

연결화(Combination)

형식지에서 형식지로 변환되는 과정은 지식이 보다 검증된 객관적 지식으로 이전되는 것을 의미한다. 정보근로자들이 만든 주관적 지식들은 평판시스템, 피드백, Trackback, 링크 등을 통해 공유되고, 가치를 평가받게 된다. 이 때 주관적 지식은 구성원들의 상호작용을 통해 객관적 지식으로 변환된다.

예를 들어, 정보근로자가 특정 콘텐츠에 붙인 꼬리표는 이 과정을 통해 공동체의 꼬리표로

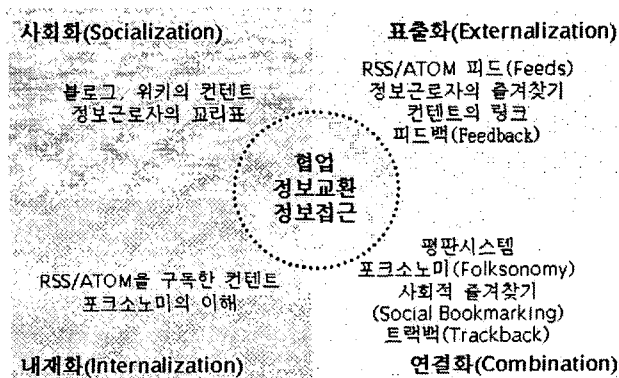


그림2 사회적 소프트웨어의 지식변환

만들어질 수 있다. 이 과정의 끊임없는 과정을 통해 포크소노미나 사회적 즐겨찾기(social bookmark) 등이 새로운 지식으로 생성된다.

위키피디아가 집단지성으로 불리는 이유는 공동체에서 인정할 수 있는 지식이 만들어 질 때까지 지식이 수정될 수 있기 때문이다. 위키피디아에 있는 지식은 공동체 구성원의 대다수가 인정할 수 있는 수준까지 수없이 변경되게 된다. 즉 검증된 객관적 지식은 끊임없는 상호작용과 재구성의 과정이라 할 수 있다. 사회적 소프트웨어는 객관적 지식의 재구성을 위해 앞서 언급한 다양한 기술과 편리성을 제공한다. 정보근로자는 자신이 스스로 생성한 지식의 변환 과정과 함께 사회적 지식의 변환 과정도 함께 관찰하고 직접 참여할 수 있다.

내재화(Internalization)

내재화는 검증된 객관적 지식을 자신의 주관적인 지식 구조로 변형시키는 과정을 말한다. 정보근로자는 공동체에서 인정된 지식을 흡수하고 기존의 지식을 활용할 수 있게 된다. 정보근로자는 RSS 구독기를 통해 공동체의 지식을 얻을 수 있다. 이 때 RSS 구독에 포함되는 항목은 개인의 지식에 직접적인 영향을 미치는 것만이 해당될 수 있다. 연결화 과정에서 협업적 필터링은 내재화를 위한 객관적 판단 기준이 될 수 있다.

결론

정보근로자는 사회적 소프트웨어를 사용하여 보다 쉽게 다른 구성원과 협업할 수 있다. 사회적 소프트웨어는 순환적 지식관리 프로세스의 각 단계에서 정보근로자가 지식화 작업을 할 수 있도록 지원한다. 알려진 바와 같이 사회적 소프트웨어는 웹을 기반으로 하고 있으며, 웹2.0과 밀접한 관계를 갖고 있다. 따라서 최근 급속히 증가되고 있는 웹2.0 서비스나 도구는 정보근로자의 지식화 과정을 보다 효과적으로 지원할 수 있다.

본 연구에서는 기업에서 사용되는 지식관리시스템과 사회적 소프트웨어의 관계나 통합 가능성은 언급하지 않고 있다. 조직 시스템으로 사회적 소프트웨어는 조직 구성원의 정보 생산성을 높일 수 있고, 기존 시스템과 달리 가볍우며(light weight), 기업 시스템에 쉽게 이식할 (embedded) 수 있다.

그러나 보안 및 시스템 통합과 관련된 이슈가 문제로 남아 있다. 따라서 기업 차원에서 어떻게 사회적 소프트웨어를 적용하고 사용할 수 있는지에 대한 연구가 요구된다. 한편 사회적 소프트웨어는 현재 급속히 발전하며 다양한 기술과 표준안이 공개되고 있다. 향후 웹 서비스와 함께 사회적 소프트웨어의 도입을 위해 표준안과 기술 적용에 대한 사전 분석이 필요하다.

마지막으로 사회적 소프트웨어는 웹을 기반으로 있기 때문에 데스크탑 애플리케이션과 데이터 공유와 교환이 어려운 것이 현실이다. 정보근로자의 데이터와 지식을 관리하기 위해 사회적 소프트웨어를 데스크탑에 도입하는 연구도 절실히 요구된다.

참고문헌

- [1] 김홍기. (1997). 지식경영에 있어서 지식의 표현과 메타지식의 역할, 산업공학회
- [2] 홍사균. (2000). 지식흡수의 학습과정 모형 연구, 과학기술정책연구, 연구보고 000-04
- [3] 조상철. (2005). 최근 업계의 Communication & Collaboration 트렌드, 마이크로소프트 비즈니스 포커스
- [4] 최호찬. (2004). 소셜 소프트웨어, 재밌다.나로시작하는 네트워크, 월간 네트워크, <http://networker.jinbo.net/nw-news/show.php?docnbr=469>
- [5] Roll, M. (2004). Distributed KM - Improving Knowledge Worker's Productivity and Organization Knowledge Sharing with Web-based Personal Publishing, BlogTalk2.0-The European Conference on Weblogs, Vienna.
- [6] Moodthy, A.G. (2005). The Rise of Social Software: What Makes Software Social?
- [7] Wagner, C. (2004). WIKI: A Technology for Conversational Knowledge Management and Group Collaboration. Communications of the Association for Information Systems, 13:265-289
- [8] Dick, S. (2003). Knowledge Creation and the Web: Factors Indicating Why Some Intranets Succeed Where Other Fail, Knowledge and Process Management, Vol. 10, pp. 207-216
- [9] Reid, G. S, Adam, F. (2000). The Road Ahead for Knowledge Management - An AI perspective, AI Magazine, Vol. 21(4), pp. 17-40
- [10] Prentiss, R. (2005). Tags: What are They Good For?, <http://prentissriddle.com>
- [11] Robert, G. J. (2003). Emerging Technologies - Blogs and Wikis: Environments for On-line Collaboration, Language Learning & Technology, Vol. 7(2), pp. 12-16
- [12] Bush, V. (1945). As We May Think, The Atlantic Monthly, <http://www.idemployee.id.tue.nl/g.w.m.rauterberg/lecturenotes/bush-1945.pdf>
- [13] Engelbart, D.C. (1962). Augmenting Human Intellect: A Conceptual Framework,

- <http://www.bootstrap.org/augdocs/friedewald030402/augmentinghumanintellect/ahi62index.html>
- [14] Christopher, A. (2004). Tracing the Evolution of Social Software, http://www.lifewithalacrity.com/2004/10/tracing_the_evo.html
- [15] Murray, T. (1972). Delphi conferencing: Computer-Based Conferencing with Anonymity, Technological Forecasting and Social Change, Vol. 3, pp. 159-204
- [16] Peter&Trudy, J.L. (1989). Humanizing Hyperspace, The Ecology of Media
- [17] Berner-Lee, T. (1996). The World Wide Web: Past, Present and Future, <http://www.w3.org/People/Berners-Lee/1996/ppf.html>
- [18] Tim, O. (2005). What Is Web 2.0 Design Patterns and Business Models for the Next Generation of Software, <http://www.oreilynet.com/lpt/a/6228>
- [19] Collaboration Loop 웹사이트, <http://www.collaborationloop.com/>
- [20] CentralDesktop 웹사이트, <http://www.centraldesktop.com/>