IDEA GROUP PUBLISHING

701 E. Chocolate Avenue, Suite 200, Hershey PA 17033-1240, USA
Tel: 717/533-8845; Fax 717/533-8661; URL-http://www.idea-group.com

ITP4959

Collaborative Knowledge Management: Breaking the Knowledge Acquisition Bottleneck

Christian Wagner
Department of Information Systems, City University of Hong Kong, 83 Tat chee Avenue, Kowloon, Hong Kong,
iscw@cityu.edu.hk

ABSTRACT

Despite decades of research and development in knowledge management, there still exists a serious knowledge acquisition bottleneck. Neither traditional Artificial Intelligence (AI) based approaches, nor the more recent less formal knowledge management techniques have been able to overcome the knowledge acquisition challenges. This article investigates knowledge acquisition bottlenecks and proposes the use of collaborative knowledge management with wikis to overcome them.

INTRODUCTION

Ever since the development of AI and Expert Systems, there has been the promise of capturing an organization's knowledge at a large scale and making it available to the entire organization. Unfortunately, these promises did not materialize (Buchanan and Smith, 1988). While there have been several early success stories, such as American Express' Credit Advisor or Digital's Expert Configurer (XCON), attempts to acquire the broad knowledge of organizations have been less fruitful. Furthermore, even the supposed expert system successes have had their critics (Buchanan and Smith, 1989). More than a decade later, a decidedly optimistic survey by Frappaolo and Wilson (2003) finds that today still 42% of the organizational knowledge only resides in people' heads and that no more than 32% of the knowledge is available in computerized form. Obviously, knowledge acquisition is a challenge. How can we extract more of the existing knowledge from organizational sources, especially from people? And how can we manage the maintenance so as to assure that the stored knowledge is accurate and up-to-date? The article will seek to answer these questions. It is organized as follows. The next section will discuss knowledge acquisition approaches and the knowledge acquisition bottleneck. A section on collaborative knowledge management technology follows, together with a discussion of how the knowledge acquisition bottleneck can be overcome. The last section draws conclusions and reflects on further work to be done in the area.

KNOWLEDGE ACQUISITION

Approaches to Knowledge Acquisition

Organizations that try to formally (based on AI methods) acquire organizational knowledge have few available alternatives. For application areas with large amounts of transaction data, the organization can use data mining to induce rules from that data. Such data mining solutions work well for applications such as credit approval. Nevertheless, the knowledge creation effort is considerable, so that organizations frequently purchase the mined rule set and simply tweak it with their proprietary data, instead of pursuing their own data mining. Without large volumes of data, the knowledge acquisition effort has to draw on experts directly and has to elicit their knowledge in rule and fact form (or similar). With high impact projects, this can and should be done under the guidance of knowledge engineers who are trained in knowledge elicitation, formalization, and representation. Smaller projects have attempted to rely on the domain experts serving as their own knowledge engineers, using an end user approach to knowledge based system development. The latter has not been very successful. Wagner (2000; 2003) demonstrates the problems arising from end user expert system development, and the limits to its application in terms of knowledge base size and maintainability. Wagner finds end user expert systems to be often poorly structured, containing holes in the knowledge base, showing highly coupled knowledge bases that are difficult to maintain, and lacking heuristic reasoning. Yet the alternative of using knowledge engineers is infeasible when an organization seeks to capture and maintain the knowledge of the majority of its employees.

What is left then? During recent years, organizational knowledge management efforts have sought to capture knowledge in less formal ways, extending their document management and groupware systems into knowledge management systems (Holsapple and Joshi, 2002; Davenport and Prusak, 1998). This has been done for instance via better indexing methods, better search engines, and hyperlinking of content. The result has been large repositories of contextualized information, which can be considered knowledge bases.

Despite this less structured approach to knowledge organization, the challenges remain the same. When organizations try to "make sense" out of large volumes of documents in their document management systems, they usually need to seek the assistance of search engines, text mining, and automatic indexing tools, resulting in an expensive solution with limited success. Furthermore, this approach is best suited only for relatively stable and centralized knowledge bases. Users of such knowledge bases often encounter information overload, irrelevant responses, or no response to queries. Alternatively, organizations might use expert reports and "harvest" expert knowledge to capture the methods used by domain experts (Snyder and Wilson, 1998). Again, this method may be limited to point-solutions, requires considerable effort and still faces knowledge maintenance difficulties (Malhotra, 2000). Other solutions, such as corporate controlled portals (or similar hub-and-spoke) arrangements can quickly suffer from outdated knowledge and lack of maintainability (Newcombe, 2000).

Knowledge Acquisition Bottleneck

In summary, then we can describe the knowledge acquisition bottleneck as follows (Wagner, 2000; Waterman, 1986).

Narrow bandwidth. The channels that exist to convert organizational knowledge from its source (either experts or documents, or transactions) are relatively narrow. Knowledge engineers can only focus on a few key applications, but not the bulk of all organizational knowledge. Data and text mining uses are limited by cost and mining effort. End user experts are slow in capturing their own knowledge.

Acquisition latency. Together with the slow speed of acquisition often also comes a time lag between the time when knowledge (or the underlying data) is created and when the acquired knowledge becomes available to be shared. This is especially a concern in dynamic environments where knowledge changes quickly and therefore the knowledge repositories always appear outdated. This challenge is both related to the methods of knowledge acquisition as well as incentive systems, which often do not encourage experts to freely share their most recent knowledge.

Knowledge Inaccuracy. Experts make mistakes and so do data mining technologies. Furthermore, maintenance can introduce inaccuracies or inconsistencies into previously correct knowledge bases. With little available bandwidth to create new knowledge, there will be frequently fewer resources available to check the accuracy of knowledge already in the system. Furthermore, correction procedures can be difficult and cumbersome (Who is permitted to correct errors? What is the procedure? What incentives are there to report errors?).

Maintenance Trap. As the knowledge in the knowledge base grows, so does the requirement for maintenance. Furthermore, previous updates that were made with insufficient care ("hacks") will accumulate and will make future maintenance more difficult (Land, 2002). Thus, the success of building an increasingly larger knowledge base, creates the trap of growing maintenance requirement that paralyzes future growth, as documented by Brooks in the realm of software (1995).

Given these challenges it appears there are few opportunities for breaking the knowledge acquisition bottleneck. Nevertheless, the next section will propose one possible remedy.

COLLABORATIVE KNOWLEDGE MANAGEMENT WITH WIKIS

A promising new area of end user developed knowledge solution via collaborative knowledge management technologies, especially with the use of wiki technology. Collaborative knowledge management means that many people work together to create or acquire knowledge, instead of a few individual experts. In other words, a community (of practice) will jointly create and maintain the knowledge. Research elsewhere (Cheung et al., 2004) suggests that *conversational* knowledge management is well suited for this challenge, whereby conversations, i.e. questions and answers, become the source of relevant knowledge.

This form of knowledge creation and management has become popular in communities that form around discussion boards. Leading solutions such as Yahoo Groups are used by Millions of communities. Yet while discussion forums have been a simple and quite practical solution, they lack several useful knowledge representation and maintenance features. A newer technology, which incorporates many of those desirable features, is the wiki. Within this section, wiki technology and its suitability for knowledge management will be discussed.

Wiki Structure and Function

A wiki is a set of linked web pages, created through the incremental development by a group of collaborating users (Leuf and Cunningham, 1999), as well as the software used to manage the set of web pages. The first wiki was developed by Ward Cunningham in 1995 as the PortlandPatternRepository, in order to communicate specifications for software design. The term wiki (from the Hawaiian *wikiwiki* meaning "fast") gives reference to the speed with which content can be created with a wiki. According to the Wikipedia (www.wikipedia.org), an online encyclopedia written as a wiki, wiki key characteristics are as follows:

- It enables web documents to be authored collectively.
- It uses a simple markup scheme (usually a simplified version of HTML, although HTML is frequently permitted).
- Wiki content is not reviewed by any editor or coordinating body prior to its publication.
- New web pages are created when users create a hyperlink that points nowhere (usually simply by writing a term in "CamelCase", concatenating two or more words and capitalizing them)

Creating Wiki Pages

Creating and editing wiki pages is a simple activity. A wiki author will use a web-enabled *formfield* to enter a comment he or she wishes to publish. Authors can use plain text or a simplified mark-up language. The system then automatically creates and publishes a web page with a unique URL that can be indexed and linked to. Hence users with virtually no web publishing knowledge can create web content as quickly as they can write a text document.

Linking Wiki Pages

A fundamental aspect of knowledge management with wikis is the use of hyperlinks. Hyperlinks link topics and create context. Wikis are designed to drastically simplify hyperlinking. Hence users do not have to create and use URLs (although they can), instead they normally use "CamelCase" (multiple words capitalized and concatenated) to create a link. Links whose destination (page) does not exist are depicted as question marks, as if the author were asking a question. Another author (or the original creator) can then respond by clicking on the question mark, thus invoking an editor to write the new page. Upon completion of the edit, the question mark will automatically convert into a hyperlink (now underlined text) to the new page.

Versioning

As multi-user systems, wikis are designed to enable any user to modify any other user's web pages (unless specifically limited by access right settings). This creates numerous challenges in version management. Wikis solve these challenges by keeping prior versions of any web page in memory, and enabling rollback, comparison, difference identification, and similar capabilities, if so desired. Furthermore, there will be a history of prior changes with author, date, and related information, as well as potentially a change explanation.

Relevant Opensource and Extreme Programming Principles

Opensource Principles

Knowledge management using wikis bears considerable resemblance to opensource software development (Markus et al., 2000), including the following traits:

- Mutually reinforcing motivations, such as sharing in the collec tive success,
- Work product open to the public and therefore easy to monitor,
- Reliance on the voluntary efforts of multiple distributed partici pants to make enhancements,
- Self-governance of the developer team,
- Task Decomposition for more development efficiency,
- Use of technology for communication and coordination and norms on how to use it.

In opensource software development communities, these traits result in ultimately lower error rates (compared to closed source), development of large applications, very fast development, and high maintainability (open source).

Extreme Programming Principles

Extreme programming (Beck, 1999) is a software development discipline designed to speed up the development process through simplification, principled approaches, systematic testing, and development tools. It is targeted for small developer teams and quick development in rapidly-changing environments. Extreme programming applies 12 principles, 9 of which with apparently high relevance to collaborative knowledge creation using wikis:

Simple, minimalist design. Only the code needed to achieve the user's requirements is created, not "nice-to-have" features or code to satisfy future requirements.

Common, shared metaphors. All developers use the same names and descriptions to communicate concerning the software.

Small releases. The software is built with small, incremental, and quickly-deployed improvements.

Refactoring. A process of software restructuring is routinely exercised whereby functionality does not change, but which simplifies the code in order to facilitate future modifications.

Testing. A testing regimen is carried out while the software is developed, with several levels of testing and supported through automated testing tools.

Pair programming. Pairs of programmers write all code.

Collective ownership. Every line of code belongs to every programmer working on the project. Hence all programmers are responsible to keep the code functional.

Continuous integration. The project advances through frequent "build" processes, possibly several times per day, creating new working versions that incorporate new features.

Coding standard. Joint standards for coding are followed by all programmers. Hence, different parts of the code are void of programmer "signatures" and joint ownership and maintainability is promoted.

The three additional, less relevant principles shall be mentioned here only for the sake of completeness, namely regular workweek, onsite customer presence to communicate requirements, and a planning process carried out jointly with users.

If we want to extract from the main principles three *meta-principles*, they should include (1) simplicity of design and frequent redesign (refactoring) to maintain simplicity, (2) frequent creation of a small work product available for review and testing, and (3) work in teams where development becomes a conversation, to facilitate back-up, clarity, and shared understanding. All of these are potentially highly relevant in collaborative knowledge management as the Wikipedia example in a subsequent section will illustrate.

Promise of Opensource and Extreme Programming Principles

Opensource software development has had remarkable successes, creating software that appears to break longstanding rules of software evolution (Scacchi, 2003). For example, opensource software size has been shown to grow super-linear (exponential), rather than linear or inverse-square.

Similarly, extreme programming offers the promise of successful and fast development of large work products, by keeping them small, simple, and easily maintainable. According to Jones (1996) a growth in the size of the work product (i.e., function points) leads to rapidly growing failure rates (project cancellation), and schedule overruns under traditional development conditions. Furthermore, traditional development approaches lead to high maintenance costs for change requests late in the development cycle (or maintenance).

Wiki Performance Evidence: Wikipedia Example

Wikipedia, the online encyclopedia developed as a wiki, adopts end embraces key concepts of opensource development and extreme programming.

First of all, the Wikipedia gains development advantages through its structure as a wiki. The combination of ease and speed of publishing content together with the ability of engage a potentially large group into the knowledge creation process, enables it to become a large and up-to-date knowledge repository. As of August 2003, the Wikipedia contains over 147,000 articles.

Wikipedia articles are updated so frequently, that major events are reflected in them typically within 24 hours. The Wikipedia also reports on historical events the day they happen (e.g., indictment of prominent figures). The history information of active Wikipedia entries often reveals multiple modifications on the same day and dozens of modifications within a month.

Figure 1: Growth in Wikipedia Article Volume Since Inception in 1/2001 (taken from http ://www.wikipedia.org /wiki/Wikipedia:Size_of_Wikipedia)

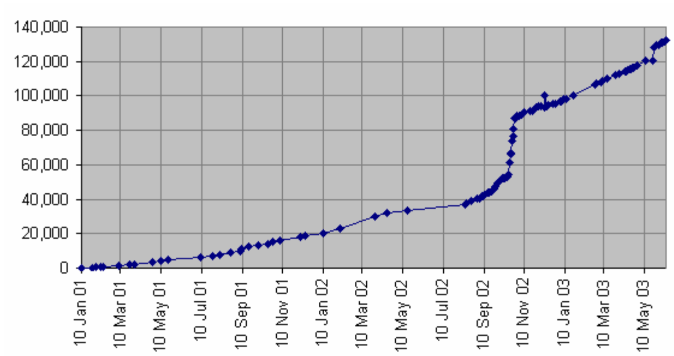

Figure 1 depicts the growth rate of the Wikipedia since January 2001. The figure shows that the first 20,000 Wikipedia entries required one year's time (10,000 pages per 6 months). Within the next year, about 80,000 pages were added (10,000 pages per 1.5 months). Since then, the growth rate is about 10,000 pages per month, suggesting overall a super-linear growth.

The "secret" of this growth may be attributed to the adherence of several principles of opensource development and extreme programming, which are highlighted below.

First, the Wikipedia can draw in a large developer group (reportedly thousands of contributors). Anyone can add content to the Wikipedia (even without registration). Any reader may take the existence of a "hole" in the encyclopedia as an invitation to contribute. Consequently, as the readership grows (Wikipedia is heavily spidered by Google and receives very favorable placement in search results), the potential author number also increases.

Second, Wikipedia pages are highly decoupled from each other, so that new authors can write with little concern for the current content on other pages.

Third, when authors make a contribution, whether writing a new page or changing a page, the result is immediately visible to the entire community, thus enabling "testing". Prior versions are also stored, so comparison and possible roll-back is easy.

Fourth, there is no individual ownership of Wikipedia pages, which are developed by volunteers, thus everyone works to improve everyone's contributions. Quality is everyone's responsibility.

Fifth, the Wikipedia has strong editing guidelines, motivated by the refactoring rules of software development. This ensures that articles which have had multiple contributors and commentators ultimately become very readable again.

As a result, Wikipedia, in only two years is challenging the otherwise largest, but "closed authorship" encyclopedia (Britannica) for leadership in content. Encyclopedia Britannica has about 85,000 entries but their average size is about double that of the Wikipedia.

Since wiki technology is relatively new and somewhat contrary to many organizations' knowledge management cultures, there are very few success cases such as the Wikipedia. However, we should expect that in the near future, an increasing number of wiki software products will emerge, and more and more communities will replace their "inferior" discussion boards with wiki technology.

CONCLUSIONS

The message of this article, put into few words, is as follows. To deal with the challenge of capturing and maintaining the possibly exponentially growing volume of knowledge requires new ways of knowledge acquisition and a knowledge management approach that relies on the contributions of many, rather than the expertise of a few. Wiki technology, and the "wiki way" of collaboration shows a feasible model for knowledge acquisition and maintenance. The Wikipedia offers an illustration of the effectiveness of this approach.

Nevertheless, knowledge acquisition and maintenance capability needs to be measured empirically to justify the claims made here. While the article suggests for instance that knowledge in a wiki can grow in super-linear fashion, this must be empirically tested, together with the wiki's response other challenges identified earlier, such as knowledge latency and accuracy. Hence, collaborative knowledge management using wikis will be a promising application for the practice of knowledge management as well as a rich source of interesting research questions.

REFERENCES

Beck, K. (1999). *Extreme Programming Explained: Embrace Change*, Reading, MA: Addison-Wesley.

Brooks F. P. (1995), *The Mythical Man-Month - Essays On Software Engineering*, 20th Anniversary Edition, Addison-Wesley Longman.

Buchanan, B. and Smith, R. (1988). "Fundamentals of expert systems", *Annual Review of Computer Science*, 3 (7), 23-58.

Buchanan, B.G., and Smith R.G. (1989). "Fundamentals of Expert Systems", in Barr, A., Cohen P.R., and Feigenbaum E.A. (Eds.), *The Handbook of Artificial Intelligence*, Volume IV, Reading, MA: Addison-Wesley, 149-192.

Cheung, K.S.K. Lee, F.S.L. Ip, R.K.F., Wagner, C. (2004) "The Development of Successful On-Line Communities", forthcoming in the *International Journal of the Computer, the Internet and Management*, 12 (1).

Davenport T. and Prusak L. (1998). Working Knowledge. *How Organizations Manage What They Know*. Harvard Business School Press.

Frappaolo, C. and Wilson, L.T. (2003). "After the Gold Rush: Harvesting Corporate Knowledge Resources". *intelligentKM*, available at http://www.intelligentkm.com/feature/feat1.shtml.

Holsapple, C.W. and Joshi, K.D. (2002) "Knowledge Management: A Threefold Framework", *The Information Society*, 18, 47–64.

Jones, C. (1996). *Patterns of Software Systems Failure and Success*, Boston, MA: International Thomson Computer Press.

Land, R. (2002) "Software Deterioration and Maintainability: A Model Proposal". *Proceedings, Second Conference on Software Engineering Research and Practise in Sweden (SERPS) Karlskrona, Sweden*.

Leuf, B. and Cunningham, W. (2001) *The Wiki Way: Collaboration and Sharing on the Internet*, Reading, MA: Addison-Wesley.

Malhotra, Y. (2000). "From Information Management to Knowledge Management: Beyond the 'Hi-Tech Hidebound' Systems." In Srikantaiah, K. and Koenig, M.E.D. (Eds.), *Knowledge Management for the Information Professional*. Medford, N.J.: Information Today Inc., 37-61.

Markus, M.L., Manville, B., and Agres, C. (2000). "What Makes a Virtual Organization Work—Lessons from the Open Source World", *Sloan Management Review*, 42 (1), 13-26.

Newcombe, T. (2000). "Opening The Knowledge Portal", *Government Technology*, available at http://www.govtech.net/magazine/gt/2000/sept/knowledge/portal.phtml.

Scacchi (2003). "Understanding Open Source Software Evolution: Applying, Breaking and Rethinking the Laws of Software Evolution", Working Paper, Institute for Software Research, UC Irvine.

Snyder, C.A. and Wilson, L.T. (1998). "The Process of Knowledge Harvesting: A Key to Knowledge Management", *Proceedings, 8th Annual BIT Conference*, Manchester, UK.

Wagner, C. (2003). "Knowledge Management through End User Developed Expert Systems: Potential and Limitations", in Mahmood, M.A. (Ed.), *Advanced Topics in End User Computing*, Idea-Group Publishing.

Wagner, C. (2000). "End-Users as Expert System Developers," *Journal of End User Computing*, 12 (3), 3-13.

Waterman, D. A. (1986). *A Guide to Expert Systems*. Reading: Addison-Wesley.