

# **The Knowledge Warehouse: Reusing Knowledge Components**

Michael Yacci  
Associate Professor  
Information Technology  
Rochester Institute of Technology  
Rochester, NY 14623  
(716) 475-5416  
email: [may@it.rit.edu](mailto:may@it.rit.edu)

# **The Knowledge Warehouse: Reusing Knowledge Components**

## **Abstract**

Currently, there is little knowledge reuse across training, documentation, and performance support. Knowledge-based materials developed for one purpose are not shared or reused in others. This article discusses the Knowledge Warehouse, a conceptual solution to this problem. It sketches the benefits of and limitations to the knowledge warehouse solution. It also takes the first steps towards defining a standardized classification scheme for storing knowledge components.

# The Knowledge Warehouse: Reusing Knowledge Components

## Introduction

We currently live in an information society--where information plays an unprecedented role in our personal and professional lives. Organizations realize that their corporate information assets are the commodities that set them apart from their competitors. Peter F. Drucker (1993) stated, "Knowledge is now fast becoming the sole factor of production, sidelining both capital and labor." There is increased interest in maximizing the value of an organization's knowledge; knowledge has become one of the most (if not the most) valuable assets of the modern organization.

Most training, performance support, and documentation groups "hard-code" knowledge into material for a particular purpose. Within this context, there is a large amount of *non-reusable* knowledge and information. This article explains the background of the knowledge reuse problem with an emphasis on training and performance support issues. It offers a conceptual solution in the notion of the knowledge warehouse, and provides preliminary guidelines for the development of knowledge components to be used in a knowledge warehouse.

## The Issues in Training and Performance

The estimated price for finished, professional video is approximately \$4,000 per final edited minute. In spite of this high cost, the value this investment is not maximized--generally such a video is not designed to be reused or shared across other related applications. This produces "brittleware" (Bassett, 1997) rather than software; the product ends up being difficult (or impossible) to reuse. In the development of most computer based training, the final product is an end to itself, rather than an information resource for the organization, largely because its components cannot be reused.

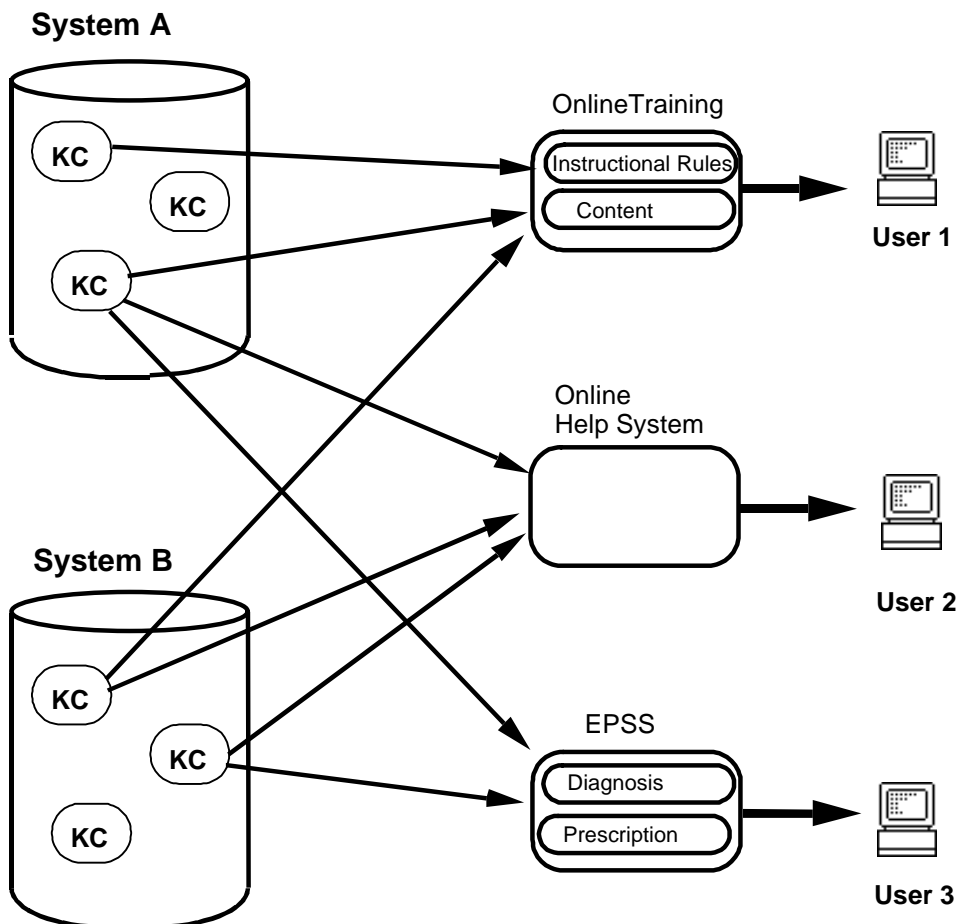
Gery (1994) suggests that human performance materials should embody "best current practice" because of the dynamic nature of the content. This implies that as processes and procedures for performance change, materials should be revised and brought up to date. Materials generally need replacement due to factors such as changing government regulations, new company policies, changed or improved technologies, or process reengineering.

This allows us to phrase a two-pronged problem: (1) "one-shot" materials in which reuse would be desirable and (2) resources with little shelf life, in which updates are critical.

## The Knowledge Warehouse

The solution to these issues may lie in the development of a *knowledge warehouse*. A knowledge warehouse can be thought of as an "information repository" in which knowledge components are cataloged and stored for reuse. A knowledge warehouse enables a variety of different *views* of knowledge, useful in areas such as training or documentation. These views could be pre-set and organized by instructional designers or technical writers. Additionally, the knowledge warehouse could also support ad hoc queries, such as electronic performance support systems, intelligent help, or reference materials. Not incidentally, knowledge can be stored in several physical places, although that is not a requirement.

The knowledge warehouse consists of knowledge components (KCs) that are defined as the smallest level in which knowledge can be decomposed. Figure 1 shows an abstract system in which knowledge components are stored in different physical locations (Systems A and B). Knowledge components from each system are assembled into on-line training or instruction for User 1. Some of the knowledge components are shared by an on-line help system, that has been assembled for User 2. User 3 is being guided by a performance support system that is analyzing the work that he performs and is using instructional/performance rules to determine which knowledge might be most useful to him at the current time.



**Figure 1.** Knowledge Components (KC) are shared across three different applications.

Similar ideas were described by Mackay (1988) who used the term *information database*. Koulopoulos & Frappaolo (1995) called a similar idea an *electronic data management system*. Gery (1991) refers to a similar idea as an *infobase*. The knowledge warehouse described here is meant to unify some of these abstract ideas into a practical system. Hence it is a conceptual amalgam of these ideas.

A knowledge warehouse parallels the recent idea of a *data warehouse*. A data warehouse is a repository for (usually corporate) data, that enables ad hoc queries and sophisticated decision support analysis (Bischoff & Alexander, 1997). Within a data warehouse, data can physically reside on any number of computers. The data warehouse software "cleans" the data so that it can be shared across computers (Dhar & Stein, 1997), handling issues of software compatibility. This permits information to be aggregated and analyzed. A major economic benefit of any database lies in storing information once, in a form that is accessible by other systems that need it. A database can produce pre-defined "views" of data, as in the case of departmental reports. A data warehouse can also support ad hoc queries to enable executives and decision makers to get at the data they need (Barquin, 1997).

### **Data, Information, and Knowledge**

The distinctions between data, information, and knowledge as technical terms are somewhat tenuous, but it is important to attempt to clarify the meanings at this point.

**Data** consist of the measurement and "computerization of daily life." (Cabena, Hadjinian, Stadler, Verhees, & Zanasi, 1998). Data might be thought of as the atoms of knowledge. Data are typically what we attempt to gather and measure, such as age, size, or amount. Data by themselves explain very little—data are the substance by which explanations are formed.

**Information** is "that which leads to understanding." (Wurman, 1990). Generally, information is considered to be organized and sorted data, that can be used for answering a specific question. Information is the aggregation and subsequent reduction of data, such as averages, trends, and percentages.

**Knowledge** may be thought of as information in use, or the set of rules and relationships that enable value added, skilled performance. Knowledge may consist of work procedures and processes, precedents, details and conceptual relationships between topics in a domain. (Klein, 1992). Knowledge is often represented in the form of an expert's rules, although we now know that rules alone generally cannot produce expert behavior (Holland, Holyoak, Nisbett, & Thagard, 1986). Knowledge is a higher level aggregation and interpretation of information and data.

Huang, Lee, and Wang (1999) also acknowledge the difficulty in defining these terms, but suggest the following:

“Data are collected, sorted, grouped, analyzed and interpreted. When data are processed in this manner, they become information. Information contains substance and purpose. “Knowledge” is generated when information is combined with context and experience.” (Page 146).

### **Other Parallels**

One last parallel between database ideas and knowledge warehousing should be considered. Data mining is the process of searching through existing data in an attempt to discover previously unknown relationships in the data. Similarly, Huang, Lee, and Wang describe *knowledge mining* as a process that parallels data mining. In essence, knowledge mining would be an attempt to find previously unknown knowledge by uncovering barely visible relationships in knowledge. This idea is described by Senge (1990; Raybould, 1997) as the *learning organization*. This evolving area of knowledge management is concerned with collaboratively learning from one’s mistakes: “lessons learned” from successful or unsuccessful projects.

While there are many issues in the realization of a knowledge warehouse, a first important issue is the size and scope of knowledge components: at what granularity should knowledge be decomposed so that it is greater than data and information, but not overly "compiled." A second issue revolves around the actual development of a working prototype of a knowledge warehouse: where is knowledge stored, how is it located, how is it represented, and how is it re-assembled. A third issue lies in the development of proper client-side control systems, so that (in the case of training) instructional presentation rules are applied to the knowledge content to produce instruction rather than reference materials. Yet a fourth issue lies in knowledge mining leading to value added knowledge creation.

The rest of this paper discusses the first issue: the size and scope of knowledge components. Discussion of other issues is deferred for future articles.

### **Knowledge Granularity**

A standardized system of classification for knowledge components (KC) is necessary to allow knowledge reuse within and between organizations. Such a classification system should decompose knowledge at an appropriate level of granularity so that knowledge can eventually be recombined. As a criterion for such a classification system, a knowledge component must maintain some aspect of knowledge rather than merely data. Therefore, it cannot be decomposed to the point that we are dealing with data. While ad hoc or proprietary classification systems can be created (using "tag" languages such as SGML and XML), our premise is that a standardized classification scheme can be evolved that will simplify the search and recombination of knowledge components.

Another look at databases might serve us well at this point. Databases are often developed by reasoning backward from *reports* (specific user views) to determine ways of structuring

the database (Kroenke, 1998). For example, if users request an employee's name, age, and years of service, it suggests that those elements should be treated as data elements. Additional reports that use only some of those elements help to confirm that they should be treated discretely, as separate pieces of data.

Following the idea of reasoning backward from a user's view, we might decompose a manual for a service technician whose job it is to maintain and repair a piece of equipment. This technician's view might include product specifications, flowcharts, diagrams, hints, detailed explanations, and definitions. A training course might use definitions, explanation, practice exercises, simulations, and some of the flowcharts. But the training may re-order the material such that it is appropriate for a novice, or may supply practice and transfer materials that are not appropriate for documentation. This suggests that these are the types of knowledge components--larger than data--that need to be considered.

### **Theoretical Perspectives**

There are several theories and instructional models that might be used to give insight into knowledge components used in instruction. Several popular models include Merrill's Component Display theory (Merrill, 1983, 1988), Gagne and Briggs's events of instruction (Gagne & Briggs, 1979) or Experiential Learning theory (McCarthy, 1987; Kolb, 1984; Yacci, 1991). Robert Horn's early work in information mapping (Horn, 1989; Romiszowski 1986) produced a classification system for information types that went beyond instruction. Horn also developed 34 types of information elements that are commonly included in technical writing and instruction. Space precludes a complete review of these models in this article.

Useful elements in the design of instruction, (using a combination of terms from several of the above sources) might be generalities, examples, explanations, practice items, test items, overviews, advance organizers, and analogies among others. A brief definition of each is given in Figure 2.

<p><b>Generality:</b> a statement or diagram that applies to all instances, such as a definition of a concept, or a flowchart of a procedure</p> <p><b>Example:</b> a specific instance of a concept, procedure, or principle</p> <p><b>Explanation:</b> a series of statements that justifies why something works, or why things are done a particular way</p> <p><b>Practice Item:</b> an opportunity for a learner to perform a task with the goal of building skill by receiving feedback on performance</p> <p><b>Test Item:</b> an opportunity for a learner to perform a task with the goal of showing competency on the task</p> <p><b>Course Maps:</b> a verbal or graphic organizer that reveals a learner's place in a given set of content</p>
--

**Advance Organizer:** verbal or graphic information presented at a higher level of abstraction than content that is forthcoming

**Analogy:** a comparison of two objects that states their similarities and differences

**Figure 2.** Some proposed knowledge components drawn from instructional design theories.

### **An Example**

Let's examine an example of an instructional view that uses knowledge components from a knowledge warehouse. An instructional view might include a variety of *lesson templates* such as: concept lessons, procedural lessons, process lessons, principles lessons, or fact-based lessons. The following hypothetical example of a completed lesson template might be a concept lesson on classifying a "legal check" from an introductory banking training session. (The following example describes the instructional view and how it is assembled.)

An instructional view would include a generality (i.e., a definition) of a legal check, listing the attributes that must be present for a check to be legal (such as signature, date, amount, account number). If a "legal check/definition" already exists as a knowledge component it can be linked or embedded in the instructional view. A set of examples is also needed, that can show several variations of acceptable and unacceptable legal checks. Existing examples of acceptable and unacceptable legal checks can be pulled from the knowledge warehouse and used in the training session. Additional examples may also be used as practice items or test items.

If such knowledge components do not exist, they can be created, tagged, and added to the knowledge warehouse for potential reuse. The *definition* knowledge component would be related to the final course by means of a relational binding, sometimes called object linking. This means that if the component is changed in the warehouse, then all views that are linked to it will also change. Note that should the definition ever be updated--if the law or the organization should require something additional in the definition of a legal check--the instructional view would automatically update. This seems to make concrete the notion of capturing "best current practice" in all views.

### **“Boiler Plate” Instruction?**

The rules by which one best combines instructional components into instructional systems has been a “holy grail” of instructional design researchers for several decades, perhaps beginning with the programmed instruction approach of the 50’s and 60’s (Klaus, 1965). A variety of instructional design models mentioned earlier, such as component display theory (Merrill, 1983), Robert Mager’s (1987) series of instructional design courses, and Gagne’s events of instruction and learning hierarchies (1979), all suggest standard elements and content sequences that should be included to develop lesson templates. More



recently, the intelligent tutoring community has similarly reached out for the same set of presentation rules as has a constructivist learning movement (Gott, Lesgold, and Kane, 1996). Left unanswered (at this point), then, is the “best” method for combining these knowledge components. It would appear that many knowledge components could be flexibly combined to produce lesson templates that conformed to many of these existing approaches.

Ideally, instructional presentation rules could be stored within a knowledge warehouse to produce instruction (more or less) automatically. Other types of presentation rules could be stored as well, such as rules that prescribe effective performance support or rules for conforming to an organization’s technical writing templates. Additionally, the final form that these knowledge components take could be hand crafted if desired. Hand craftsmanship dilutes some of the cost effectiveness of a knowledge warehouse system, but might produce more effective materials in some cases. The effectiveness of “hand crafted” vs. “mass produced according to stored rules” vs. “standard template” instructional sequences would require testing.

### **Advantages**

Cost savings from knowledge reuse could be enormous. The development of Computer Based Training (CBT) was estimated in the early 90's as 400 person hours to one hour of finished CBT. While new software tools may enable some improvement on these numbers, the cost of developing CBT from scratch--without reuse-- is still high. Because these ideas are in a prototyping phase, there is currently no empirical data to support the cost of developing such a knowledge warehouse. However, using cost benefit data from the software engineering word suggests that reusability of components can significantly reduced development costs.

Corporations that have practiced reuse of software components in the area of software engineering have gained up to 50% productivity increases, up to 84% lower project costs, cycle time reductions of 20-70% and 20-30% reduction in defects (Poulin, 1997). Similar gains in productivity in the training and documentation areas could be expected with the onset of knowledge warehouses and reusable knowledge components.

### **Limitations**

The success of a knowledge warehouse depends upon developers designing for reuse. That means that knowledge developers must not presuppose the final implementation of the knowledge (Wlodarczyk, 1998). This in turn requires careful isolation and classification of knowledge components. Each must be treated like a black box that has an input and an output (or perhaps a *start* and a *finish*). There must be little overlap between components so that they can be assembled later into a variety of views.

One of the current limitations of a knowledge warehouse is the current state of database software. Currently, digital video and audio require a substantial storage capacity.

However, since courses would be *views* of the knowledge warehouse rather than finished products, knowledge components can be stored in a variety of locations, and need only come together on a desktop. Hence a single storage capacity is not necessary, (but might be considered for other reasons). While we are a long way still from the "paperless office" concept, we move closer to the day when knowledge does not take a physical form--and instead exists solely electronically. This allows knowledge to be transformed and reassembled for different purposes via intranets or the internet at very little cost.

Certainly, courses could be temporarily "frozen" and downloaded to local machines to improve delivery speed. In the short term this might be necessary because of speed issues of downloading rich graphics and high bandwidth video. However, these may not always be issues. Additionally, these may not be issues for "in-house" training courses that never leave the corporate backbone.

### **Conclusion**

A knowledge warehouse is an inevitable move towards treating the intellectual work of an organization as an asset to be managed. It requires that knowledge management become more systematic and predictable. The technology to accomplish knowledge warehousing is currently available; this article takes a first step towards defining knowledge components.

## **References:**

- Barquin, R.C. (1997). A Data Warehousing Manifesto. In R. Barquin and H. Edelstein (Eds.), *Planning and Designing the Data Warehouse*. Upper Saddle River, NJ: Prentice Hall.
- Bassett, P.G. (1997). *Framing Software Reuse*. New Jersey: Yourdon Press Computing Series
- Bischoff, J. & Alexander, T. (1997). *Data Warehouse: Practical Advice from the Experts*. Englewood Cliffs, NJ: Prentice Hall.
- Cabena, P., Hadjinian, P., Stadler, R., Verhees, J., & Zanasi, A. (1998). *Discovering Data Mining*. Upper Saddle River, NJ: Prentice Hall PTR
- Dhar, V. & Stein, R. (1997). *Intelligent Decision Support Methods*. Englewood Cliffs, NJ: Prentice Hall.
- Drucker, P.F. (1993). *Post-Capitalist Society*. New York: HarperBusiness
- Gagne, R.M. & Briggs, L.J. (1979). *Principles of Instructional Design*. New York: Holt, Rinehart and Winston.
- Gery, G.J. (1991). *Electronic Performance Support Systems*. Boston: Weingarten.
- Gery, G.J. (1994). *Performance Support: A Gold Mine for Training*. ASTD Satellite Videoconference, Indianapolis Indiana. September 28, 1994.
- Gott, S.P., Lesgold, A, & Kane, R.S. (1996). *Tutoring for Transfer of Technical Competence*. In B.G. Wilson (Ed.), *Constructivist Learning Environments*. Englewood Cliffs, NJ: Educational Technology Publications.
- Holland, J.H., Holyoak, K.J., Nisbett, R.E., & Thagard, P.R. (1986). *Induction: Processes of Inference, Learning and Discovery*. Cambridge, Massachusetts: MIT Press.

- Horn, R.E. (1989). Mapping Hypertext. Lexington, MA: The Lexington Institute.
- Huang, K., Lee, Y.W., Wang, R.Y. (1999). Quality Information and Knowledge. Upper Saddle River, NJ: Prentice Hall PTR.
- Klaus, D.J. (1965). An Analysis of Programing Techniques. In, R. Glaser (Ed.), Teaching Machines and Prograded Learning, II. Washington, D.C.: National Education Association.
- Klein, G.A. (1992). Using Knowledge Engineeringto Preserve Corporate Memory. In, R.R. Hoffman (Ed.) The Psychology of Expertise. New York: Springer-Verlag.
- Kolb, D.A. (1984). Experiential Learning. Englewood Cliffs, NH: Prentice-Hall
- Koulopoulos, T.M. & Frappaolo, C. (1995). Electronic Data Management Systems: A Portable Consultant. McGraw-Hill.
- Kroenke, D.M. (1998) Database Processing: Fundamentals, Design, and Implementation (6th Edition). Englewood Cliffs, NJ: Prentice Hall.
- Mackay, W. (1988). Tutoring, Information Databases, and Iterative Design. In D.H Jonassen (Ed.), Instruction Designs for Microcomputer Courseware. Hillsdale, NJ: Lawrence Erlbaum Associates.
- Mager, R.F. (1988). Making Instruction Work. Belmont, CA: David S. Lake Publishers.
- McCarthy, B. (1987). The 4Mat System. Barrington, IL: Excel.
- Merrill, M.D. (1983). Component Display Theory. In C.M. Reigeluth (Ed.), Instructional Design Theories and Models: An Overview of their Current Status. Hillsdale, NJ: Lawrence Erlbaum Associates.
- Merrill, M.D. (1988). Applying Component Display Theory to the Design of Courseware. In D.H Jonassen (Ed.), Instruction Designs for Microcomputer Courseware. Hillsdale, NJ: Lawrence Erlbaum Associates.
- Poulin, J.S. (1997). Measuring Software Reuse. Reading Massachusetts: Addison Wesley, Inc.
- Raybould, B. (1997). Performance Support Engineering: An Emerging Development Methodology for Enabling Organizational Learning. Performance Improvement Quarterly, 10(1) pp. 167-182.
- Romiszowski, A.J. (1986). Developing Auto-Instructional Materials. London: Kogan Page.

Senge, P.M. (1990). *The Fifth Discipline*. New York: Doubleday/Currency.

Włodarczyk, P. (1998). Personal communication.

Wurman, R.S. (1990). *Information Anxiety*. New York: Bantam Books.

Yacci, M. (1991) CBT Applications of Kolb's Experiential Learning Theory. Paper presented at The 33rd International ADCIS Conference, St. Louis Missouri.

**About the Author:**

Michael Yacci, Ph.D.  
Associate Professor  
Information Technology  
Rochester Institute of Technology  
Rochester, NY 14623  
(716) 475-5416

email: may@it.rit.edu

Ph.D. Instructional Design, Development, and Evaluation. Syracuse University.

I am interested in all aspects of performance support systems. I am also interested in using information technology and artificial intelligence techniques to create more cost-effective instructional materials. I have been working on several prototype projects in these areas.

I am currently a Consulting Editor for *Performance Improvement Quarterly* and am the Book Review Editor for *IEEE Software*.

**Interests:**

Intelligent Tutoring Systems  
Information Technology and Instruction  
Performance Support Systems