Paper 144-27

The Knowledge Warehouse: The Next Step Beyond the Data Warehouse

Anthony Dymond, Dymond and Associates, LLC, Concord, CA

ABSTRACT

Retaining and applying knowledge enables an organization to know what to do, how to do it, and when to do it, empowering it to succeed in competitive situations. Data warehouses store data and can be a source of knowledge but do not store knowledge directly. However, knowledge in the sense of procedures, best practices, business rules, expert knowledge, facts within context, and processed data can be retained in logical structures accessible by computers.

This paper describes the techniques and tools for building a knowledge warehouse and compares these to the data warehouse construction model. The data warehouse "extract, transform, load" (ETL) process has a parallel in the knowledge warehouse. The knowledge warehouse also has logical structures to store knowledge that are analogous to the system of tables that implement data storage in the data warehouse. Knowledge is applied through a layered representation that is readable by both humans and machines and that shields code until the bottom layer. This representation is also a system executable that is portable and can be run under SAS on a computer to help make decisions and take actions.

INTRODUCTION

Companies have spent time and money discovering important information about themselves, their customers, and their competitors. They have invested in what SAS calls "The Power to Know.™" And now that they have knowledge, what are they going to do next? How will they manage this knowledge and apply it to their business needs? The answer to that is knowledge management, and the technology to organize and store knowledge is the knowledge warehouse.

Knowledge today is viewed as one of a company's most valued assets. Once knowledge becomes available, it has to be leveraged to affect the corporate bottom line. In a sense, data warehousing and data mining are "mid-game" activities. They help to make knowledge available. Knowledge management, where knowledge is applied to the company's business activities to produce tangible results, is the "end-game." This knowledge is more than an accumulation of facts. It is also the rules and procedures needed to make decisions.

What Is Knowledge?

In the world of relational databases, knowledge takes the form of numbers or text in a data table. But knowledge is much more complex than just a recitation of processed data. It is fundamentally important to have knowledge describing business models and procedures. "Business process management" or "business rules" are often used to capture how activities are performed within an organization Best practices capture the knowledge and behaviors of skilled employees. Expert knowledge is similar, but often refers to professional knowledge. Knowledge management dealing with factual data must also consider the issue of context. This requires gathering the assumptions, prerequisites, and consequences associated with the data.

Where Do We Currently Store Knowledge?

Employees and consultants are a main repository of enterprise knowledge, but these people leave and take their knowledge with them. Documents such as procedure manuals can store rules and processes. Computer programs and databases also store knowledge, and many companies have now had decades to embed their important business processes within computer systems. Good documentation is needed to locate and understand this material and, lacking this, the true repository of business process knowledge often lies only in the program code itself.

Knowledge Management Objectives

Knowledge is an asset, and it should be enhanced like any other asset. Knowledge also usually has a half-life and has to be refreshed to retain its value. Once captured, knowledge must be retained and transferred. Even when knowledge exists in a company, there is a problem with knowing that it exists, knowing where it is located, and sharing the knowledge. The final step in knowledge management is applying the knowledge to obtain a return on investment. Figure 1 shows the goals of knowledge management.

DATA WAREHOUSE REVIEW

A data warehouse gathers data from multiple sources and organizes it in a logical and accessible structure. One of the motivations for using a data warehouse is efficiency. The original data may be located in different computers, databases, and formats. Every time a study is done, it is necessary to gather and prepare data from these many sources. If the same preparation work is being done repeatedly, then it makes sense to try to find a way to do most of the preparation phase once and then reuse it.

Quality Assurance also motivates building data warehouses. Loading data into the data warehouse is a good opportunity to ensure that the data is complete and accurate. A data warehouse also provides one consistent assembled view of the data, rather than a collection of scattered files. There is less chance for confusion and misunderstanding.

Without a data warehouse, some of the data may not even be known to exist. And if it has been located, different platforms and databases may make the data almost inaccessible. A data warehouse places the data in one place in a form that is accessible to most users.

The data warehouse is organized by subject and built using tables. The central table contains detail records. The columns are the variables that experience has shown are used repeatedly to answer questions about a particular subject area. The rows represent detail activities such as individual sales. This central detail table is called a "fact table."

Some of the columns in the fact table can be joined with matching columns in "dimension tables." The dimension tables allow additional information to be joined to the detail data when needed.

The fact table and dimension tables are referred to as a "star schema." (Imagine a star with radiating lines.) This is a common structure for a subject area in a data warehouse. There may also be other tables in the warehouse such as a monthly summary table. The table structure is shown in Figure 2.

From the designer's perspective, the data warehouse starts with raw feed data that may be on different computer platforms and in different types of databases. These files must first be accessed, followed by transformation and loading into the warehouse detail table.

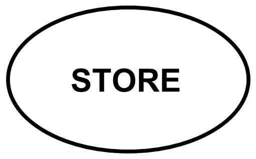
A data exploitation layer is established that can include approaches such as producing regular reports and exposing data through a browser interface. Subsets of the data warehouse can be made available as data marts.

The entire process is documented by the metadata, which is data about the data. Metadata is all of the documentation describing the data and the processes that create it. This can include both technical and business information.

KNOWLEDGE WAREHOUSE DESIGN

The data warehouse can be thought of as a three-part process of capturing, storing, and accessing the data. This three-part data warehouse "bowtie" structure is also found in the knowledge warehouse. (See Figure 3.)

Knowledge Warehouse Storage



In the data warehouse, data about a subject is stored in a set of tables. As we have designed the knowledge warehouse, the storage structure is referred to as a knowledge base and is constructed as a tree with objects at the nodes.

Figure 4 shows a knowledge base to implement a simple business rule to cross the street as:

```
IF checkLeftOK='true' AND
checkRightOK='true'
        THEN crossStreetOK='true'
```

Objects are packages containing data in "attributes" and blocks of program code in "methods." The objects are often described using a three-part box listing the object name, the attributes, and the methods. The object at the tree root node in Figure 4 is named "CrossStreetOK." If we use "dot notation" as object.attribute and object.method(), we can say that this object has an attribute

CrossStreetOK.crossStreetOK and a method CrossStreetOK.isCrossStreetOK().

We would expect this method to use the attributes CheckLeftOK.checkLeftOK and CheckRightOK.checkRightOK from the descendant objects to calculate a value for the attribute CrossStreetOK.crossStreetOK. Similarly, the methods CheckLeftOK.isCheckLeftOK() and CheckRightOK.isCheckRightOK() contain the code that will produce values for the attributes in these terminal nodes.

The methods are activated by way of tree search algorithms. These algorithms are procedures to transverse the branches of the tree. As the "focus of attention" of an algorithm touches each node, the method code will run. For example, a depth-first tree search algorithm starting at the root node will transverse the tree in the order:

1. CrossStreetOK
2. CheckLeftOK
3. CrossStreetOK
4. CheckRightOK
5. CrossStreetOK

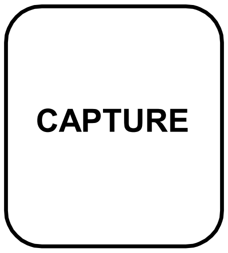
This tree search supports a control structure. Assume when the process reaches the object CheckLeftOK that it finds CheckLeftOK.checkLeftOK='false.' The search next returns to CrossStreetOK where CrossStreetOK.crossStreetOK can now be proven false. The search process can be stopped since it is no longer necessary to execute methods in CheckRightOK.

Identifying and placing objects as nodes in the tree, followed by identifying the attributes and methods, accounts for two-thirds of the knowledge base design effort. All that remains is writing the methods code, which in this system can be done using the Base SAS language.

The tree and search algorithms capture the functional, hierarchical, and temporal relationships between objects and provide this in an understandable visual presentation. Relocating much of the control structure into the tree greatly reduces the amount of control code in the methods and allows fast development of smaller and more maintainable systems.

Our original goals included capturing business process, procedure, and rules. Examination of Figure 4 shows the complete automation of a process to cross the street. Our goals also included capture of best practices and domain expertise. Again, viewed from this perspective, Figure 4 shows the encapsulation of domain knowledge of how to cross a street and directly supports an expert consultation system.

Knowledge Warehouse Capture



Knowledge bases are frequently entered into the knowledge warehouse by project teams conducting object-oriented analysis and design (Figure 5). Team analysis and design methodologies such as the "unified process" can be helpful and are well described in the literature (Larman, 2002).

Several automated techniques can help in converting data into a form that is loadable into the knowledge warehouse. For example, rules can be generated directly from the association statistics used in market basket analysis. By noting co-occurrence rates for scarves, hats, and gloves worn by people entering a building on a winter day, it is possible to determine association probabilities and produce a rule as:

```
IF hat AND scarf
       THEN gloves (p=0.60)
```

Note that this rule suggests an association but not cause and effect.

Team analysis and statistical discovery are not mutually exclusive. This same rule can also result from the design team’s insights and analysis.

Another mechanism to expand a knowledge base is to find a new case that resembles an existing case. For example, a new animal might closely resemble a horse except for having striped hair and living only in Africa. A knowledge base could extend itself by taking the object Horse and inserting an edited version into the knowledge base as the new animal Zebra. This is referred to as "case-based reasoning" or "memory-based reasoning."

New data can also be discovered in decision trees built by data mining and then remapped into the knowledge base.

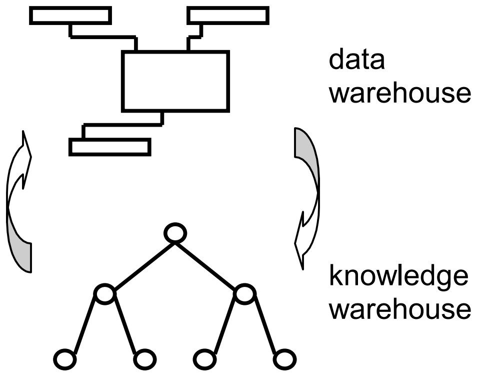
Knowledge Warehouse Access

The knowledge warehouse can be accessed by executing stored knowledge. It can also serve as the repository to document a company's business knowledge. Since the knowledge warehouse contains a great deal of information encapsulated in attributes and methods, it can be dynamically queried against. Finally, the knowledge bases, objects, attributes, and methods are themselves components that can be reassembled into new structures (Figure 6).

Executable Knowledge

It is possible to build and execute object models of processes and procedures. The CrossStreet knowledge base allowed executing an object model of a business process built around business rules.

Applying intelligent data warehousing (IDW), it is possible to have the object model read and write against a data warehouse and its metadata as the process executes. This allows processes to dynamically use metadata, build run-time predictive models from data warehouse tables, and write new rows to the data warehouse.



Documented Knowledge

Important enterprise knowledge is no longer stored in documents, databases, computer programs, and people’s heads. It is stored in a knowledge warehouse that is accessible to both humans and computers. The issue of separate and desynchronized programs and documentation is mostly resolved because the programs and documentation are in the same structure.

The business user interacts with knowledge captured in a visual format, enabling a flexible, easily customized system that can respond quickly to changing business requirements. Applications can be built much more quickly than by using conventional programming, and the application can be understood and often maintained in the business units themselves without the overhead and delays associated with IT support.

Knowledge Queries

The knowledge base's attributes and methods can be made accessible to complex queries. For example, a knowledge base to identify animals could also support a query against attributes for habitat and food sources to answer questions about resource competition.

Additional methods can be constructed to support special queries. For example, the information about hats, scarves, and gloves that evolved the rule:

```
IF hat AND scarf
       THEN gloves (p=0.60)
```

can be rearranged to produce a method of the form:

```
(hat, scarf) = placeNextTo(gloves)
```

Building this method will allow a query that returns a list of items that gloves should be placed next to in a store.

Reassembling Knowledge

Executable knowledge is modularized and ready to use in support of enterprise requirements. The knowledge is available at different granularities for reassembly into new applications.

Documented and run-time accessible knowledge structures exist from individual knowledge base, objects, methods, and attributes. Reassembly into new knowledge bases is often little more than a cut and paste process. Knowledge bases can also dynamically load and execute other knowledge bases, allowing *ad hoc* navigation through the global knowledge space. Multiply entrant tree structures and flexible search algorithms enhance knowledge access.

CONCLUSION

Knowledge–knowing what to do, how to do it, and when to do it–gives an enterprise a competitive advantage. The knowledge warehouse, as an extension of the data warehouse, provides a mechanism to capture, store, and access knowledge. Knowledge can be captured by a combination of team-centered techniques, such as the "unified process," and by automated techniques such as are found in data mining. Knowledge can be stored in a tree structure with software objects placed at the tree nodes. Combined with tree search algorithms and tree manipulation methods, this provides a powerful built-in control structure at the system level. Knowledge bases in the knowledge warehouse can provide a software object model of a business process with embedded intelligence. They can dynamically interact with a data warehouse while executing an enterprise's knowledge.

REFERENCES

Dymond, A.M. (2000), "An Object-Oriented Expert System for the SAS Environment," *Proceedings of the Eighth Annual Western Users of SAS Software Regional Users Group Conference*, Scottsdale, Arizona.

Dymond, A.M. (2001), “Telling Aardvarks from Zebras with an Expert System Written in SAS,” *Proceedings of the Twenty-Sixth Annual SAS Users Group International Conference*, Long Beach, California. (CDROM paper 135-26).

Dymond, A.M. (2001), "Knowledge Management Using an Expert System Written in SAS," *Proceedings of the Southeast Regional Users Group Conference*, New Orleans, Louisiana.

Dymond, A.M. (2001), "Intelligent Agents and Applications in Enterprise Computing," *Proceedings of the Western Regional Users of SAS Software Ninth Annual Conference*, San Francisco, California.

Jackson, P. (1999), *Introduction to Expert Systems*, Harlow, England: Addison Wesley Longman Limited.

Larman, C. (2002), *Applying UML and Patterns. An Introduction to Object-Oriented Analysis and Design and the Unified Process (2nd Edition)*, Prentice-Hall.

O'Dell, C.S., and C.J. Grayson (1998), *If Only We Knew What We Know: The Transfer of Internal Knowledge and Best Practice*, Free Press.

Russell, S., and P. Norvig (1995), *Artificial Intelligence: A Modern Approach*, Prentice-Hall.

Stewart, T.A. (1997), I*ntellectual Capital: The New Wealth of Organizations*, Doubleday.

ACKNOWLEDGEMENTS

Automated Consultant is a trademark of Dymond and Associates, LLC.

SAS and all other SAS Institute Inc. product or service names are registered trademarks or trademarks of SAS Institute Inc. in the USA and other countries. ® indicates USA registration.

Other brand and product names are registered trademarks or trademarks of their respective companies.

CONTACT INFORMATION

Anthony M. Dymond, Ph.D.
Dymond and Associates, LLC
4417 Catalpa Ct.
Concord, CA 94521

(925) 798-0129
(925) 680-1312 (FAX)
amdymond@dymondassoc.com

A SAS Institute Inc. Alliance Partner

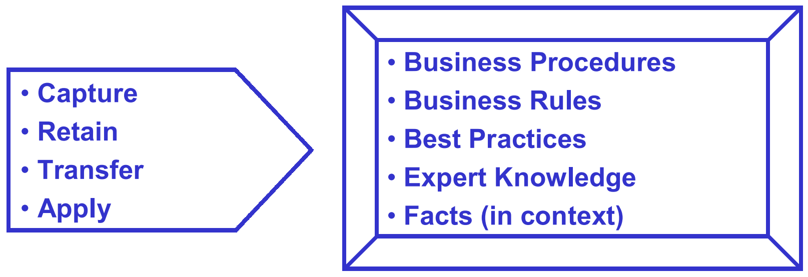


Figure 1: Goals of a Knowledge Management System

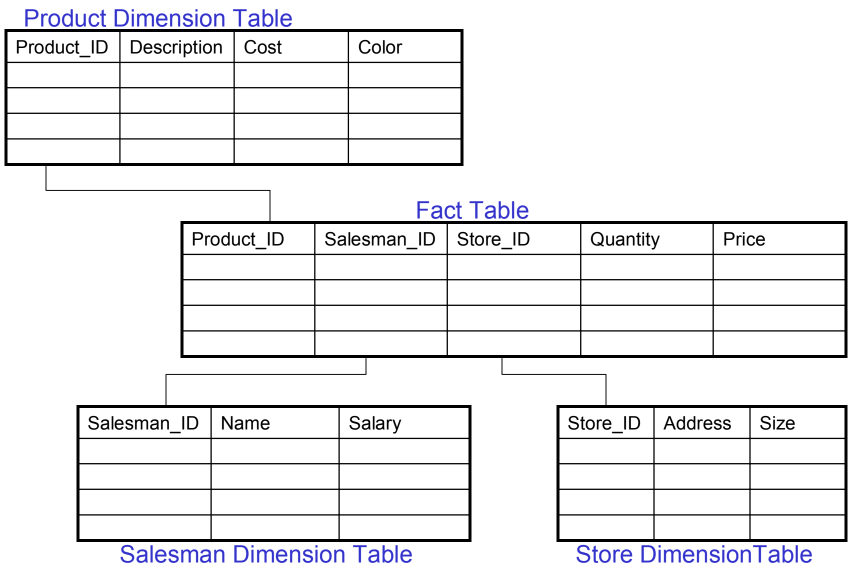


Figure 2: A Data Warehouse Stores Data in Tables

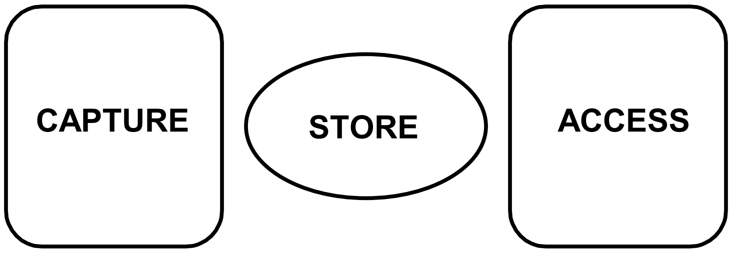


Figure 3: The Data Warehouse and the Knowledge Warehouse Capture, Store, and Provide Access to Information

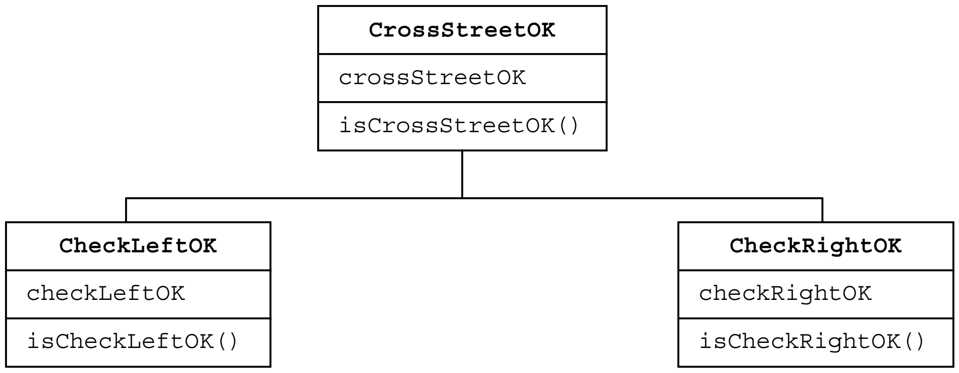


Figure 4: Objects and Tree for a Knowledge Base to Cross the Street

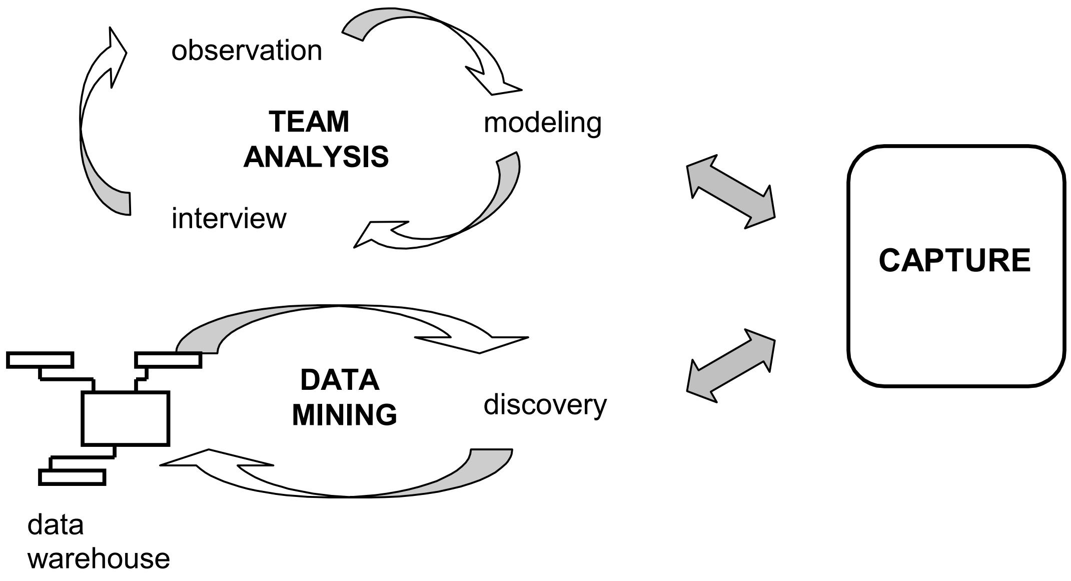


Figure 5: Capturing Knowledge for the Knowledge Warehouse

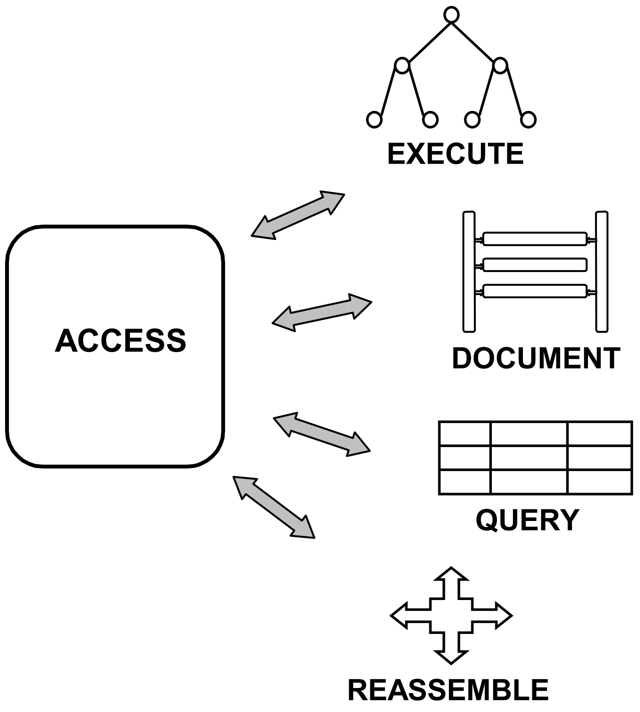


Figure 6: Accessing Knowledge from the Knowledge Warehouse