

# Knowledge Management in Enterprise Resource Planning Systems

Daniel E. O'Leary  
University of Southern California  
3660 Trousdale Parkway  
Los Angeles, CA 90089-1421

[Oleary@rcf.usc.edu](mailto:Oleary@rcf.usc.edu)

From: AAI Technical Report WS-99-10. Compilation copyright © 1999, AAI (www.aaai.org). All rights reserved.

## Abstract

The purpose of this paper is to investigate the development of knowledge management systems designed to support enterprise resource planning systems, such as SAP. Knowledge management can be used in virtually every facet of supporting ERP system choice, design, development, testing, supporting, and reporting. This paper summarizes a number of actual examples and discusses some emerging efforts, focusing on knowledge management, with particular interest in case-based knowledge management.

## Introduction

One of the most rapidly growing areas of software implementation is what is referred to as "Enterprise Resource Planning" (ERP) Systems. Perhaps the best known of the ERP systems include those known as the "big five," SAP, PeopleSoft, Oracle Applications, Baan and J. D. Edwards.

ERP systems provide firms with software that integrates functional areas by focusing on processes. ERP systems provide transaction processing capabilities that help to integrate a firm's information systems. Typically, ERP systems are based on a relational database. Using a relational database and appropriate process design, allows the firm to capture data once and then make that data available for use throughout the firm, by all appropriate users.

ERP systems have been well-received. It has been estimated that virtually all of the Fortune 500 firms have either implemented an ERP system or are implementing an ERP system. Implementation of ERP systems has grown to be huge business. It has been estimated that roughly 50% of the consulting done by the major consulting firms has to do with choosing, designing, developing, testing or implemented ERP systems. As a result, there is a large potential for knowledge-based support of these systems.

ERP systems are large. Quantum's implementation of Oracle's ERP application reportedly has over 40,000 tables. In addition, increasingly, ERP implementations are accompanied by large data warehouses, designed to facilitate data access.

One of the emerging phenomena is the development of knowledge management systems to support choosing, designing, developing, testing, implementing, using and reporting for ERP systems. The purpose of this paper is to discuss some of these knowledge management system developments, with particular emphasis on the potentially increasing role of case-based reasoning in those knowledge management systems.

## Choosing ERP Systems

The need for knowledge management systems to facilitate the choice of ERP systems was illustrated in Timberjack's recent choice of an ERP system (Romanow et al. 1998). Timberjack was faced with trying to find a system that met both the needs of their USA and Finnish offices. Sharing information was an important vehicle in exchanging ideas on which software vendor to choose. Unfortunately, much sharing was done using paper. As a result, not all the critical knowledge necessary to facilitate decision making was readily available to all in a timely manner. For example, an important article was made available only to the Finnish office regarding a European ERP implementation.

A knowledge management system designed to capture and make available information regarding the various ERP systems in a timely manner would have been a helpful device. Such a system could have included both generic information related to each ERP system and firm specific developments, such as requirements and other issues.

## **Implementing ERP Systems: Support for Implementation Difficulties**

Developing ERP systems requires that both users and developers coordinate their efforts. As users find problems they need to make the developers aware of those problems and developers need to find solutions. In some cases users may have the same or a similar problem. If the problem has been solved, then users need to be able to find out about the previous solution. If the problem has not yet been solved, but others have the same problem, then there needs to be a coordination of efforts, because otherwise resources will be misallocated with the development of duplicate solutions.

### **Case Study**

As part of the development of an ERP, a consultant had developed a knowledge base that would capture of problems and allow tracking of their solution (O'Leary 1999a). As noted by the consultant during an interview

(The system is used to)...categorize problems by issue category, e.g., training, system, network, PeopleSoft Configuration Set Up, PeopleSoft Module, etc. ... a user logs in an issue with information, such as which module, process, screen shot, ... (that is the source of the problem). Then, a consultant opens it to solve the problem. ... Any progress or resolution to an issue will be logged into this database. ... This way ... the project can have better resource allocation planning.

Originally, the system was developed in order to provide a quick fix to track an overwhelming number of user support requests. In addition, there had been duplicate inquiries that ultimately led to redundant efforts, that the knowledge management system now minimized. However, the system has been so useful that the client planned to continue to use the system. Future plans included changing the computing environment, migrating the system to Lotus Notes.

This knowledge base is now being extended. A case-based system is now being designed, couching the data as cases in order to more fully exploit machine processing capabilities of the cases.

### **ERP Reporting Systems**

Unfortunately (fortunately?), reporting systems for ERP systems are generally perceived as difficult to use. Starting with Microsoft's implementation of SAP, there

has been a push to make ERP report information available on corporate intranets (O'Leary 1999b). As a result, ERP report information increasingly is being treated as part of firms' knowledge management systems.

In the case of Microsoft, expert users were expected to use the reporting capabilities of the ERP system. However, for less expert and casual users, a wide range of information generated from the ERP was made available on the intranet. In addition, to making the information widely available, the move of information to the intranet minimized ERP costs that Microsoft incurred, since those costs were based on a per seat level of usage.

### **Developing Data for Input to ERP Systems: Support for Financial Transactions**

Using ERP systems requires that the user be aware not only how to use the system, but also understand what they want to do with the system. In order to get users the knowledge that they need to do their job, a number of firms have made "help" available over the intranet.

### **Financial Transactions**

ERP financial modules require that users be able to provide transaction information for use in the system. Unfortunately, transactions vary in their difficulty. I interviewed some financial system users that indicate that development of transactions for entry into the system can be very difficult. In some cases, only experts are able to develop those entries. As a result, there is interest in developing system support for those users. For example, recently I had a conversation with a representative of one of the big five ERP firms who indicated that their firm was interested in obtaining knowledge-based support for those personnel making financial system transactions.

Unfortunately, there has been limited research to-date on formal knowledge representation of transaction knowledge to guide development efforts. Perhaps the only paper to model the knowledge-based representation of financial transactions is O'Leary and Kandelin (1992). They developed a domain specific natural language-based system that understood accounting language and was able to use that understanding of events in order to generate the resulting financial transaction entries.

### **Transaction Characteristics**

An important characteristic of financial transactions is that they tend to repeat themselves. For example, an overwhelming majority of a firm's financial transactions are purchases or sales. Within these two types of transactions there is substantial similarity. The basic form

of all purchases is roughly the same, including either a cash disbursement and a purchase or an accounts payable and a purchase. "Purchases" is a generic account sometimes replaced by an account representing the specific item purchases, such as materials or a particular kind of materials. This repeating nature leads us to suggest that a case-based reasoning approach is an appropriate vehicle to capture and represent knowledge about financial transactions.

Although transactions basically "repeat themselves" they are not always identical. Case-based reasoning is necessary because, for example, previously stored financial transactions may need to be modified to make new transactions and/or new transactions does not match 100% any of the previously stored transactions and thus a selection must be made based on partial match.

### **A Prototype System**

Using the M.4 shell, we developed a preliminary system design that exploits the basic underlying database requirements for financial transactions (see exhibit 1 for a sample case and a few rules). In particular, research suggests that ERP systems employ a database schema that generates information on resources, events, agents and locations from the financial transactions (O'Leary, 1998). Case attribute information is captured based primarily on the events for which the system is designed to process. Additional information in the cases includes the resource (e.g., cash) and the direction of change in the resource (e.g., cash increasing), the external agent (e.g., the particular client) and the location for which the event is occurring (e.g., central office). In addition, the resulting financial system entry, in terms of debits and credits also is captured.

### **Data Warehouses for ERP Systems: Creating Knowledge**

Increasingly, firms are making ERP information on their intranets. However, typically, ERP-based reports are designed for a single month, quarter, or year. As a result, some users are now interested in longer time frames of available data. As a result, firms are implementing data warehouses to facilitate access to a broader range of the data over longer time periods, such as multiple years.

Data warehouses open up additional opportunities for knowledge management and case-based reasoning. In particular, knowledge discovery can be used to exploit the extensive data availability in a data warehouse.

### **Extensions**

This paper has focused on knowledge management for choosing, implementing, using and reporting in ERP systems. However, knowledge management systems could also be developed for other aspects of ERP systems, such as, designing, developing and testing.

A prototype system design was presented for capturing case-based knowledge about financial transactions. The case base in support of that system is minimal and could be further extended.

### **Summary**

Enterprise resource planning (ERP) systems are receiving widespread attention. Knowledge management systems are being developed to facilitate use of those ERP systems. This paper has investigated some of the relationships between ERP systems and knowledge management systems, with a focus on the opportunities for use of case-based knowledge representations.

### **References**

- O'Leary, D., "SAP is REAL," paper presented at SMAP 98, New Orleans, August 1998.
- O'Leary, D., "James O'Leary Travel Services, III: A Second PeopleSoft Implementation," Unpublished Case Study, University of Southern California, 1999a.
- O'Leary, D., "Microsoft's Reporting in SAP," Unpublished Case Study, University of Southern California, 1999b.
- O'Leary, D., and Kandelin, N., "Accountant: A Domain Dependent Accounting Language Processing System," in *Expert Systems in Finance*, D. O'Leary and P. Watkins, 1992, Elsevier.
- Romanow, D., Keil, M. and McFarlan, W., "Timberjack Parts: Packaged Software Selection Project," Harvard Business School, February 25, 1998.