Knowledge Management in Enterprise Resource Planning Systems: Support for Financial Transactions

Daniel E. O'Leary

University of Southern California
3660 Trousdale Parkway
Los Angeles, CA 90089-1421

Oleary@rcf.usc.edu

Abstract

ERP system users have indicated a need for support for users responsible for generating financial entries. This paper presents a prototype case-based system designed to support use of an enterprise resource planning system's financial entries. This kind of a system could be linked to users through an ERP portal access.

1. Introduction

Enterprise resource planning (ERP) system financial modules require that users be able to provide transaction information. Unfortunately, for a number of reasons users need support in the development of transactions. First, transactions vary in their difficulty. Interviews with 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 correctly to develop those entries. Second, in some cases, "substitute" users take responsibility for transaction entry. "Original" users may be on vacation or ill. As a result, users may not be adequately prepared to place financial entries into the system. Third, ERP systems are complex and their use is not easy. Substantial training and user support is necessary to facilitate system access and use.

As a result, there is interest in developing system support for those users. For example, recently a representative of one of the leading ERP firms indicated that their firm was interested in obtaining knowledge-based support for those personnel making financial system transactions. Accordingly, the purpose of this paper is to investigate the design of such a system and review a prototype version of a system designed to assist financial users in the development of transactions.

Previous Research

Unfortunately, there has been limited research to-date on formal knowledge representation of financial 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.

Plan of this Paper

This paper proceeds in the following manner. Section 1 has provided an introduction to the importance of this problem, stated the purpose of this paper and briefly summarized the previous research. Section 2 provides a brief summary of ERP systems. Section 3 discusses the case-based nature of financial transaction information and knowledge. Section 4 discusses the system design and prototype implementation. Section 5 examines some extensions. Section 6 summarizes the paper.

2. Enterprise Resource Planning Systems

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 ERP firms" or BOPSE firms: **B**aan, **O**racle Applications, **P**eopleSoft, **S**AP (the largest ERP firm) and J. D. **E**dwards.

ERP systems provide firms with software that integrates multiple functional areas, focusing on processes, rather than function. ERP systems provide transaction processing capabilities that help to integrate a firm's information systems. Typically, ERP systems employ 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. As a result, there is a large potential for knowledge-based support of these systems.

3. Financial Transaction Characteristics

An important characteristic of financial transactions is that they tend to repeat themselves. First, some transactions occur at regular intervals over time. Second, other events frequently occur. For example, an overwhelming majority of a firm's financial transactions are purchases or sales. Third, even transactions that are not frequently occurring, may repeat themselves, if only occasionally. 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.

Because transactions repeat themselves, capturing previous transactions and reusing those transactions can provide support to financial system users. This capability suggests that previous entries could be used to guide user construction of financial entries.

Although transactions basically "repeat themselves" they are not always identical, even if it is only the dollar amounts that change. As a result, case-based information about financial transactions is not intended to replace the user, but only provide support.

4. A Prototype System

Using the M.4 shell, a preliminary system design was developed that exploits the basic underlying database requirements for financial transactions (see exhibit 1 for a simple 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.

As a result, the underlying database structure is used as the basis of the cases. In particular, the Resources – Events – Agents – Location (REA) is used as the basis of capturing information about the entries for the case library. REA (McCarthy 1982) provides a theoretical structure to accounting and other database applications.

The initial and driving matching criteria for the cases in the system are the events (E), 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. Although REA does not require a debit – credit structure, ERP systems still employ that approach. As a result, the system is designed around a debit – credit structure.

5. Extensions

This paper has focused on supporting ERP system users in the area of financial transactions using case-based reasoning. Additional system support could be generated to support users of other modules.

A prototype system design, based on REA, was presented for capturing case-based knowledge about financial transactions. That approach centered on the event of concern. At this point, the system has been developed as a proof of concept. As a result, the system's case base and approach is minimal and could be extended.

In addition, ultimately for such a system to be used as part of an ERP system would require integration into the ERP system or into the portals that are being generated by ERP vendors. Clearly either approach is outside the scope of this paper.

One approach would be to embed the system into an ERP portal (e.g., figure 1). Similar to other knowledge support that is available from the portal, a simple virtual tab or link could tie the portal to the transaction system support system. Using this approach, such a system could be loosely linked, rather than tightly integrated into the ERP.

6. Summary

Enterprise resource planning (ERP) systems are receiving widespread adoption. Unfortunately, these systems are complex. As a result, systems are being developed to facilitate use of those ERP systems. In particular, there has been concern by ERP firms and client firms about supporting users of different modules. This paper has investigated the opportunity of supporting financial transactions. A prototype system that employed an REA structure was developed and discussed. A number of extensions were examined, including the potential for linkage of such a system to users through an ERP portal.

References

McCarthy, W., "The REA Accounting Model: A Generalized Framework for Accounting Systems Shared Data Environment," **Accounting Review**, July 1982, pp. 554-577.

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.

Figure 1

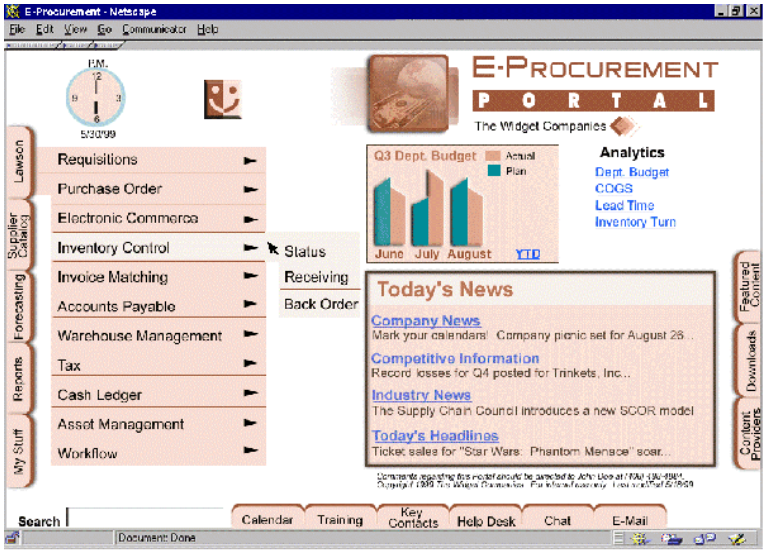

Exhibit 1

```
/*------------------------ object class definition -----------------------*/

classdef(entries) = [supers = []].

/*------------------------ class instance of entry event cases -----*/

instdef(sales_event) = [
  class = entries,
  event_category = sales_event,
  resource_increasing = cash,
  location = central_office,
  external_agents_impacted = customers,
  debit = cash,
  credit = sales ].
/*-----------sample goal, questions, legal values and rules--------------*/

goal = [final_conclusion].

question(event_category) = 'What category of entry are you considering?'.

legalvals(event_category) = [sales_event, purchase_event].

question(status) = 'Would you like to continue considering case attributes?'.

legalvals(status) = [continue, all_done].

if event_category = Category and
   classinst(entries, ENTRIES) and
   ENTRIES <- getslot(event_category) = Category and
   ENTRIES <- getslot(resource_increasing) = C and
   display(["Found a resource increasing-based match for ",C,nl])
then conclusion_match = good.

if event_category = Category and
   classinst(entries, ENTRIES) and
   ENTRIES <- getslot(event_category) = Category and
   ENTRIES <- getslot(event_category) = D and
   display(["for a ",D,nl])
then event_conclusion = good.

if conclusion_match = good and event_conclusion = good and status = all_done then final_conclusion =
good_match.
```