Chapter 15

Component-based ERP Design in a Distributed Object Environment

Bonn-Oh Kim
Seattle University, USA

Ted Lee
Memphis State University, USA

ERP (Enterprise Resource Planning) vendors have seen a dramatic increase in their sales this decade. Even though several vendors are producing great products and making huge profits, there are some problems to be resolved to make ERP applications a continuous success in the next decades. Current ERP applications have the low reusability and interchangeability of various modules among different vendors’ packages. One of the main reasons for these shortfalls is a tight coupling of ERP domain knowledge with the particular implementation tools. Also, efforts in establishing and using the standards in specifications of ERP applications have been inconsequential. In this article, strategic steps to wield a dominant power in the future ERP market are discussed. These steps are as follows: 1. Knowledge Modeling: Abstraction of Domain Knowledge from Tools; 2. Componentization of Domain Knowledge; 3. Implementation of Componentized Domain Knowledge; 4. Marketing Strategies for Domain Knowledge Components.

INTRODUCTION

Since the early 1990s, a notion of business reengineering has been very popular in many companies, especially in the USA. One of the contributions of business reengineering is that corporate information systems should be viewed as an enabler to transform the business processes and consequently organizational struc-
tures. To fulfill the mission of an enabler of business transformation, corporate executives found that corporate information systems should be planned, designed and implemented from an enterprise-wide perspective. A collection of islands of software located in various divisions of an organization could not satisfy the new needs of large corporations.

To deliver an integrated set of software systems for various functions of a company, including accounting, manufacturing, logistics and others. Recently, ERP vendors such as SAP, Baan, PeopleSoft, Oracle and J. D. Edwards have seen their sales growing exponentially. Behind the successful stories of ERP, however, there are several issues to be dealt with in order to adapt to the ever-changing computing environment and maintain the competitive advantages.

Borrowing the idea from the industrial manufacturing, software components built based on standard specifications can be a building block for resolving the current problems in designing ERP applications. To build software components, however, we need to have a set of specifications at the knowledge level. In this article, knowledge modeling abstracted from the implementation tools is discussed as a precursor for building the components for ERP applications after the problems of current ERP applications are discussed and core competencies of ERP vendors are reviewed from a perspective of overall computing architectures.

PROBLEMS OF CURRENT ERP DESIGN

Currently, each ERP vendor has been developing its own proprietary systems in various domain areas. Since ERP customers prefer the seamless systems across their business functions, ERP vendors are continuously expanding into new domain areas. However, one vendor does not necessarily produce superior ERP packages across all business functions. Each vendor maintains superiority in some functional domains, e.g., PeopleSoft for human resource management.

From a perspective of ERP customers, they have to opt for using all ERP applications primarily from one vendor or selecting many packages from different vendors. If customers can choose the best from different ERP vendors without worrying about the compatibility among different vendors’ ERP packages, they can maximize the productivity gains by installing the best ERP applications in their organization. From an ERP vendor’s perspective, it is very difficult to specialize in any particular domain functions (e.g., manufacturing, financials, etc.) because many customers want a smorgasbord of ERP packages from one vendor. If ERP packages from different vendors are interchangeable or compatible, some problems aforementioned can be somewhat resolved.

There have been overlaps in efforts developing virtually the same type of applications (e.g., accounting packages) by many different vendors. Reinventing a wheel is a last thing we need to do. Current ERP designs in industry lack the
Related Content

Fault-Prone Module Prediction Approaches Using Identifiers in Source Code
[www.igi-global.com/article/fault-prone-module-prediction-approaches-using-identifiers-in-source-code/121546?camid=4v1a](www.igi-global.com/article/fault-prone-module-prediction-approaches-using-identifiers-in-source-code/121546?camid=4v1a)

Natural Language Processing: An Inevitable Step in Requirements Engineering
[www.igi-global.com/article/natural-language-processing/104655?camid=4v1a](www.igi-global.com/article/natural-language-processing/104655?camid=4v1a)

LOCALE: Collaborative Localization Estimation for Sparse Mobile Sensor Networks
[www.igi-global.com/chapter/locale-collaborative-localization-estimation-sparse/66476?camid=4v1a](www.igi-global.com/chapter/locale-collaborative-localization-estimation-sparse/66476?camid=4v1a)
An Empirical Study on Novice Programmer’s Behaviors with Analysis of Keystrokes
www.igi-global.com/article/empirical-study-novice-programmer-behaviors/77619?camid=4v1a