Context-Aware Framework for ERP

Farhad Daneshgar
University of New South Wales, Australia

INTRODUCTION

Like many existing ERP models (e.g., Podolsky, 1998; Van Stijn & Wensley, 2001), the OOAB framework is also based on a widely accepted assumption that a corporate-wide information system consists of a set of potentially related subsystems; and as a result, information flows among these subsystems must be identified, and required resources planned, using an appropriate ERP methodology. However, up until now there existed no formalised framework that facilitates sharing of contextual knowledge in ERP processes. A unique attribute of the OOAB framework is that it treats ERP processes as a collaborative processes where various roles/actors collaboratively perform tasks in order to achieve a common overall goal. An object-oriented framework is presented in this article that facilitates sharing the contextual knowledge/resources that exist within ERP processes. Context is represented by a set of relevant collaborative semantic concepts or "objects". These are the objects that are localised/contextualised to specific sub-process within the ERP process.

BACKGROUND

From a purely object orientation perspective, a collaboration is defined as "the structure of instances playing roles in a behavior and their relationships" (OMG, 2001). The behaviour mentioned in this definition refers to an operation, or a use case, or any other behavioural classifier. This article provides an overview of a framework for analysing awareness requirements of the actors in ERP systems using an object-oriented awareness-based approach. A similar study was also conducted for developing a new version of this framework that takes into consideration the specific characteristics of virtual communities (Daneshgar, 2003). The proposed approach specialises the notion of collaboration and extends it to the ERP processes. This has roots in the activity network theory (Kaptilini et al., 1995) and is based on the fact that all ERP processes involve *multiple roles performing various tasks* using appropriate artefacts (e.g., departmental sub-systems, databases, etc.) in order to achieve both their local as well as the overall organization-wide goals. Conceptually speaking, this will justify a frame-based object-oriented approach to analysis and design for ERP processes (Turban & Aaron, 2001). The conceptual model of the proposed framework is made of the following components:

- a set of collaborative semantic concepts including roles, the tasks that these roles play within the process, and the artefacts that these roles use to perform various tasks within the process, and
- relationships among these semantic concepts.

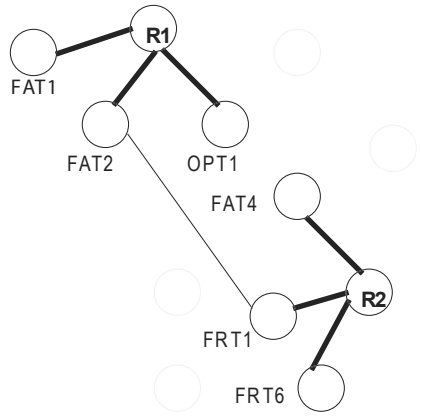
This conceptual model can then be mapped directly to an object model and be used as an analytical tool for identifying awareness requirements of the actors within the ERP process. The fact that ERP is treated as a collaborative process calls for a mechanism for maintaining awareness requirements of the actors involved in this collaboration. Furthermore, due to its object orientation, the framework is capable of encapsulating all complications and dependencies in sub/local processes within individual tasks as well as resources required to perform those tasks, further relieving the ERP management and the associated software.

OOAB FRAMEWORK

A domain-specific conceptual model of a hypothetical ERP process that resembles an object diagram is shown in Figure 1. Use of a domain-specific conceptual model instead of a straight object diagram is justified by the fact that the ontological foundation of the framework prevents growth of the objects and relationships indefinitely, and as a result using an object model may hide such ontology. In Figure 1 there are two roles: R1 and R2; six tasks: FAT1, FAT2, OPT1, FAT4, FRT1 and FRT6 (all shown by circles). It also shows various resources by straight lines connecting tasks and roles. These lines represent rich ontological relationship between a pair of semantic concepts. Each task object requires certain resources for achieving its local/departmental goal or purpose (called *process resource*), as well as certain other resources for achieving the collaborative organization-wide goals of the ERP process (called *collaborative resource*). In Figure 1, a line connecting a role vertex to a task vertex is a process resource, whereas a line connecting two tasks together is a collaborative resource.

According to the framework, effective knowledge and/or resource exchange among actors is closely related to the level of awareness as defined in the awareness model that each

Figure 1. A representation of an ERP collaborative process model



actor possess about the ERP process. These awareness levels are defined in terms of the collaborative semantic concepts used within the ERP conceptual model as shown in Figure 1. Details of the proposed methodology for identifying awareness requirements of actors in ERP process follow:

STEP 1. Develop an ERP Process Model similar to that in Figure 1.
FA: Financial Accounting sub-process/task
OP: Order Processing sub-process/task
CS: Customer Service subprocess/task
FR: Financial Reporting subprocess/task
T1…T6: <appear as postfixes indicating various tasks>

STEP 2. Measure the actual levels of awareness for each role on the process model using the awareness model. In order to measure this level of awareness the actor must be exposed to all the objects on the ERP process model, and be asked to identify those objects that s/he is aware of. Selected pool of objects are then used by an awareness model in order to arrive at a number reflecting the actual level of awareness associated with that role.

STEP 3: The actor's actual level of awareness is then is compared against the required level of awareness; the latter is a parameter, provided by the task that the actor performs within the process. The difference between these two levels of awareness constitutes the collaborative requirement of the actor for that particular task. Factors that affect the required level of awareness of a task include organisational culture, and the nature of task itself. Without possessing such awareness level the actor will not be able to collaborate with others optimally. A comparison between the actual level of awareness of the actor and the required level of awareness of the task will result in one of the following two outcomes:

1. The task's required level of awareness is either equal to, or less than, the role's actual level of awareness. This indicates that the role is qualified, or has sufficient level of awareness for taking up the task, and the OOAB framework cannot enhance collaboration any further.
2. The task's required level of awareness exceeds the role's actual level of awareness. This indicates potential for enhancing collaboration. To do so it will be necessary to put the missing objects within the focus of the actor in a way that s/he can perceive these objects, receive required awareness, and perform that particular task successfully. This will require additional resources in order to enhance the actor's awareness level. These required resources may include one or more of process resources, collaborative resources, and other communication resources, for example resources that provide awareness about other roles and other tasks within the ERP process.

IMPLEMENTATION ISSUES

One method for integration of the OOAB framework with the existing ERP systems is by developing an organisational infrastructure that provides business intelligence to the users of the ERP system by maintaining contextual knowledge that these users/actors require for effective collaboration within the ERP process. The writer is in the process of developing an expert system that provides expert advice for answering the following two specific questions:

(i) In terms of awareness requirements, is an actor capable of performing certain tasks within the ERP process?
(ii) If not, what objects need to be put within his/her focus in order to enable the actor to perform the task properly?

The ERP collaborative process of Figure 1 consists of 15 objects, including two roles, six subprocesses/tasks, six role artefacts and one task artefact. Within each of these objects is encapsulated all relevant contextual knowledge as well as pointers to relevant objects as determined by the process map. Each task possesses a set of attributes and relevant methods; and each method consists of a set of steps that corresponds to codes describing a codified knowledge. These attributes will indicate to which subprocess the task belongs to. This will enable an actor to play various roles within different subprocesses without being permanently linked to a specific subprocess, a factor that can remove some complexities in existing ERP implementations.