Chapter 7
Architecture for ERP System Integration with Heterogeneous E-Government Modules

Lars Frank
Copenhagen Business School, Denmark

ABSTRACT

ERP (Enterprise Resource Planning) systems consist normally of ERP modules managing sale, production and procurement in private businesses. ERP systems may also have modules for special lines of business or modules for the different sectors of E-Government. However, the ERP systems of today use a common database and therefore, it is normally only possible to use modules supported by the ERP supplier. This limits the possibilities for special lines of business like the different sectors of E-Government. It is normally not possible to use the traditional ACID (Atomicity, Consistency, Isolation and Durability) properties across heterogeneous ERP modules and therefore, it is not possible to integrate such modules without inconsistency and anomaly problems. That is, the users cannot trust the data they are reading and even worse they can undermine the validity of the databases if they update the databases by using such invalid information. However, it is possible to use so called relaxed ACID properties. That is, it should, from a user point of view, look as if the traditional ACID properties were implemented, and therefore, the users can trust the data they are reading and cannot do anything wrong by using this data.

INTRODUCTION

In this chapter, we will describe how the architecture of a distributed modular ERP system can be used to integrate ERP systems with heterogeneous modules specialized for E-Government.

A distributed modular ERP system is a set of ERP systems/modules that may operate in different locations where each ERP module can use the resources of the other ERP modules in its own or in other ERP locations (Frank, 2008). In a distributed modular ERP system, the ACID properties must be relaxed both across locations and across modules in the same location. That is...
within a module the traditional ACID properties of a DBMS may be used while relaxed ACID properties are implemented by the modules themselves for all accesses/updates across modules. Each ERP module has full autonomy over its own resources/tables. That is its tables may be recovered independently from the tables owned by other ERP modules.

In a distributed modular ERP system, it is easier to integrate special line of business ERP modules as the modules may be heterogeneous and have different suppliers.

In the extended transaction model used in this chapter the relaxed ACID properties are implemented in the following way: The global atomicity property is implemented by using compensatable, pivot and retrievable subtransactions in that order. The global consistency and isolation properties are managed by using countermeasures (Frank and Zahle, 1998), and the global durability property is implemented by using the durability property of the local DBMS (Data Base Management System) systems.

Applications with relaxed ACID properties function from a user point of view as if the traditional ACID properties were implemented by the underlying DBMS systems. However, in reality the relaxed ACID properties are implemented as an integrated part of the applications. Therefore, it does not matter whether all the different ERP modules use the same DBMS system or not.

The chapter is organized as follows: First, we will describe the transaction model used in this chapter. Next, we will describe the properties of a distributed modular ERP system that are necessary for integrating heterogeneous ERP modules. This is illustrated by a description of how to integrate E-Government modules in a business oriented ERP system. Finally, we have future work and conclusions.

Related Research: The transaction model described in section 2 is the countermeasure transaction model (Frank and Zahle, 1998 and Frank, 2010). This model owes many of its properties to (Garcia-Molina and Salem, 1987; Mehrotra et al., 1992; Weikum and Schek, 1992 and Zhang, 1994). The architecture of a distributed modular ERP system was first described in Frank, 2008. However, this description focused on migrating ERP systems and not in integrating traditional ERP modules and E-Government modules. Frank and Munck, 2008, describes how it is possible to integrate electronic health records from different hospitals by using relaxed ACID properties and Frank and Andersen, 2010, describes how it is possible to solve/reduce the problems in integrating incompatible tables used by heterogeneous modules.

**THE TRANSACTION MODEL**

A multidatabase is a union of local autonomous databases. Global transactions (Gray and Reuter, 1993) access data located in more than one local database. In recent years, many transaction models have been designed to integrate local databases without using a distributed DBMS. The countermeasure transaction model (Frank and Zahle, 1998) has, among other things, selected and integrated properties from these transaction models to reduce the problems caused by the missing ACID properties in a distributed database that is not managed by a distributed DBMS. In the countermeasure transaction model, a global transaction involves a root transaction (client transaction) and several single site subtransactions (server transactions). Subtransactions may be nested transactions (i.e. a subtransaction may be a parent transaction for other subtransactions). All communication with the user is managed from the root transaction, and all data is accessed through subtransactions. The following subsections will give a broad outline of how relaxed ACID properties are implemented.