

Software Technology Behind Computer Accounting

M. Župan, V. Budimir

Abstract—The main problems of data centric and open source project are large number of developers and changes of core framework. Model-View-Control (MVC) design pattern significantly improved the development and adjustments of complex projects. Entity framework as a Model layer in MVC architecture has simplified communication with the database. How often are the new technologies used and whether they have potentials for designing more efficient Enterprise Resource Planning (ERP) system that will be more suited to accountants?

Keywords—Accounting, Enterprise Resource Planning, Model-View-Control, Object Role Modeling, Open Source

I. INTRODUCTION

IN December 2009 CFO Publishing Corp. has published research on annual costs, spent by 157 US mid-sized Enterprises on modifying and updating their ERP systems. According to a source [1], 52% of the respondents estimate internal costs alone at between \$100,000 and \$500,000 per year. 43% of respondents estimate additional external costs and maintenance and support fees to be in that range. Neither time to complete change projects is not satisfied. 41% say that time to complete is worse than expected and half of all respondents say that moderate modifications take more than 20 person-days of effort. According to Paolo Juvara [2], CEO of one of the most popular open source ERP project, customers in general are not very happy with ERP complexity, inflexibility and budget overruns. Costs and complexity of modification are the key factors that should be taken into account while evaluating ERP offers. On the other hand, quality ERP software is developing on the best practice and represents optimal solution for specific business. Is it more convenient to adapt the processes and procedures of the built-in capabilities and strengths or to adapt the ERP system to own business processes? Croatian medium Enterprises size and IT budget definitely cannot be compared with the US respondents because 48% of surveyed companies have annual revenue from \$100 million to \$500 million. However, Croatian companies also have a need for ERP upgrades that comes from: security needs, legislative changes, changes in accounting methods and other financial management policies and practice, level of entrepreneurial activity or mergers and acquisition activities. Costs and complexity of ERP systems, rise of Cloud computing, Open Source licensing and n-tier architecture may result with some changes to Accounting Information Systems (AIS).

M. Ž. Author is with the Polytechnic of Požega, Republic of Croatia, 34000 Požega (phone: +385-34-211-456; e-mail: mzupan@vup.hr).

V. B. Author is with the Polytechnic of Požega, Republic of Croatia, 34000 Požega (phone: +385-34-211-456; e-mail: vbudimir@vup.hr).

II. INTEREST FOR OPEN SOURCE ERP SOLUTIONS

Open source means that code is distributed with the software and it is allowed to change it or use it. So, it does not mean free. But it is the usual practice that open source based software firms offer its community edition with no charge. Compiere, Inc., for example, has an ERP free edition, but full functionalities are available in Enterprise Edition that costs \$995 per year, per user. As Compiere stated on official web site, costs of Enterprise Cloud Edition are still 64% lower than the Microsoft midmarket ERP.

“Open” also means that system is usually developed on the open and free platforms such as Linux, using Integrated Developments Environments such as Eclipse, connected with the open databases (MySQL, PostgreSQL), what results with the lack of operating costs. Microsoft has also recognized the importance of support to OSS, although is well known by its closed, proprietary source. Microsoft supports JQuery JavaScript library within its web development tool and has even launched their ASP.NET MVC web framework under the Microsoft Public License (MS-PL). MS-PL is simple, clear, truly “open” and approved by Open Source Initiative [3].

TABLE I
 NUMBER OF DOWNLOADS FROM SOURCEFORGE.NET
 FROM JAN 1 2010 TO MARCH 3 2011

Open Source ERPs for SMBs	No.	Top OS	Top country
Openbravo (fork of Compiere)	392.777	Windows (81%)	Spain (8%)
ADempiere (fork of Compiere)	223.651	Windows (77%)	China (10%)
xTuple	143.154	Windows (62%)	USA (28%)
Opentaps	85.990	Windows (82%)	China (29%)
Compiere	81.457	Windows (84%)	China (32%)
JFire	35.762	Windows (79%)	China (21%)
OfBiz	15.624	Windows (63%)	France (60%)
TinyERP (openerp since 2008)	4.769	Windows (83%)	India (15%)
weberp	2.065	Windows (75%)	UK (10%)

Table I shows downloads rate on the world’s largest Open Source software (OSS) development website (SourceForge.net). Although downloads rate is not a key indicator of system adoption, in a combination with the global monthly searches for open source ERP systems from Table II, leads to the fact that the great interest for that kind of business

applications exists. Research is done with the help of a free Google keyword tool inside the AdWords service. Global Monthly Searches are approximate 12-month average of user queries for the keyword on Google search. Data are collected on March 31 2011, with the "Only show ideas closely related to my search terms" option and by removing queries that contain two or more different ERP names, to avoid redundancy. Keywords included in the research are names of the most popular ERPs intended primarily for the small and medium businesses, so the results are comparable with the 232.361 queries committed for one of the most popular proprietary business applications for SMBs, Microsoft Dynamics NAV.

During the research, any significant efforts for Croatian

integrates information and processes of an organization into a single database or unified system [4].

Accounting Information System is a fundamental subsystem of ERP. In the course of time, it lost its monopoly on accounting information and became integrated with other modules. Its task is collecting, processing and production of financial information.

Modules are usually parts and processes inside specific service or enterprise function, under authority of specific department [5].

As Table II shows, all OSS ERP projects are relatively new so the used development technology is pretty important. Robustness of development architecture is very important because the projects are the results of a hard work of large

TABLE II
 INTEREST AND SOFTWARE TECHNOLOGY OF THE MOST POPULAR OPEN SOURCE ERP SYSTEMS

Open Source ERPs for SMBs	Global Monthly Searches	Project started	User Interface	License	Localizations	Language	Architecture	MVC design pattern
Openerp (TinyERP until 2008)	222.808	2005	GTK+	GNU, GPL	22	Python	3-tier	+
Openbravo (fork of Compiere)	163.336	2006	Web-based	Mozilla Public License 1.1.	12	Java EE	3-tier	+
ADempiere (fork of Compiere)	57.032	2006	Web-based, Java Swing	GNU, GPL	14	Java EE	3-tier	+
OfBiz	47.404	2001	Web-based, java Swing	Apache License 2.0	N/A	Java EE	3-tier	+
Compiere	37.127	2001	Web-based, Gnome, KDE, Win32	GNU, GPL	7	Java EE	3-tier (accounting) & 2-tier the rest	+
Opentaps	19.182	2005	Web-based, Java Swing	GNU, GPL	6	Java EE	3-tier	
weberp	16.551	2011	Web-based	GNU, GPL	31	PHP		
xTuple	7.583	2007	Qt	CPALv1	6	C++	2-tier	
ERP5	5.540	2007	Web-based	GNU, GPL, Zope public license	15	Python	3-tier	
JFire	3.880	2006	Web-based, java SWT	GNU Library, LGPL	2	Java EE	5-tier	

localization of global open source ERP projects have not been observed, neither Corporate-sponsored nor independent projects.

Although academic environment, government and Association of Employers work on elimination of complex administration and business obstacles, and introduce measures to simplify starting of a business and to strengthen entrepreneurial activities, encouraging local or international open source and free software projects are not recognized as part of a strategy. That could be the result of quality and price favorable offer of Croatian ERPs that are ready to use and adjusted to the Croatian legislation.

Croatian ERPs market has only one free open source ERP solution. It is a desktop application called Spa-Erp, developed on Java platform, but without quality documentation, code comments and strong community of developer and users it is not so reliable for adjustments without parent firm.

III. SOFTWARE ARCHITECTURE OF MODERN ERP SYSTEMS

Enterprise Resource Planning (ERP) system represents the method of integrating the different functional activities of an organization into a unified structure or system by using software and hardware applications. It is a system that

number of programmers. Usually, Open Source projects are developed with the usage of OS technologies. Evolution of the Web technologies and the rise of Cloud computing [6] enabled Web-based user interface and Software as a Service (SaaS) model, which simplified usage, upgrades and maintenance. SaaS represents web application and one of the form of Cloud computing.

The term *Cloud* has been already mentioned in the 60s of the last century, representing the Internet and the world of centralized servers without human intervention [7]. But it is revitalized today as type of parallel and distributed system consisting of a collection of interconnected and virtualized computers that are dynamically provisioned and presented as one or more unified computing resources based on service-level agreements established through negotiation between the service provider and consumers [8].

Another fact that can be derived from Table II is a dominance of 3-tier technology and Model-View-Controller (MVC) design pattern. With 2009 R2 version, Microsoft NAV has also become 3-tier application. Former versions had 2-tier architecture.

IV. ACCOUNTING SETUP

The way the *posting* is solved will be described on the examples of Openbravo, one of the most popular Open Source Enterprise Resource Planning (ERP) systems, with the 2.000.000 downloads at this moment, and Microsoft Dynamics NAV, one of the most popular proprietary ERP systems. Both systems are targeted on SME sector worldwide.

Openbravo is the Spanish web-based open source ERP solutions provider, has started as a project in 2001 and received \$6.4 million in venture capital funding in 2006.

Openbravo ERP system follows 3-tier architecture and Model-View-Control (MVC) concept that enabled separate developing of data layer, business logic and user interface. All together with the modularity approach makes it easier for work of large number of developers on the same project and upgrading without changes to customize and localize modules. Wizard for Application Development (WAD) engine executes and recompiles the application every time the system administrator changes the configuration according to user request. Java code is generated by WAD, from metadata that defines windows and processes. Software design relies on metadata stored in a dictionary to model the behavior of the application, so the customization to the end users is done without much of programming knowledge [9].

Openbravo customization capabilities are proved by 59 localization projects. 12 projects can be declared as finished at this moment. Every project has its discussion forum and a leader. Localization begins with the user interface translation and importing local chart of accounts. Local taxes set-up, reports customization and creation of the specific posting schemes for every document is the second stage. Localization modules are separate from the core, so new version upgrades will not impact it.

Accounting Information System in Openbravo is called Financial Management. It is organized through the documents, like purchase order, sales invoice, settlement and cash journal. Financial Management is the base module that cooperates with Procurement, Warehouse, Project and Service, Production, Sales management modules. Most part of the documents can generate an accounting entry in the general ledger journal. It is possible to remove all documents postings, change accounting parameters and automatically re-build the general ledger journal. Posting document creates new row in a database table called *Fact_Acct* [10]. It is the base for generating financial reports. Financial report can be generated for whole organization, profit center or project, through the accounting dimensions specification.

The problem is that average user with the accounting knowledge cannot create custom accounting posting schema and connect with the document so easily. Accountant also has a need to create custom journal posting that is not connected with any document, but represents some outstanding business event. Openbravo user has "GL Posting by Database Tables" option, but the creation of DB Tables is not user friendly at all.

Despite the fact that proprietary vendor products, like Microsoft Dynamics NAV, enable user friendly adding of accounting schemas, posting setup is also rather complex and demands different approach to creating journal vouchers in a

contrast with the manual accounting.

NAV use posting groups to specify which general ledger accounts are affected with the post transactions. It consists *General posting groups* for business and for products. Business posting groups are assigned to customers and vendors. Product posting groups are assigned to items and resources. So the same item can be posted to different sales accounts in the general ledger due to customers being assigned different business posting groups. *Specific posting groups* are assigned to customers, vendors, items, bank accounts, and fixed assets. They represents direct link to the primary balance sheet account for each master item. Their purpose is to identify the primary balance sheet account for each supporting ledger. *VAT posting groups* are used to calculate Value Added Tax. They are assigned to items and resources, in a similar way as the general posting groups.

ERP systems differ in a quality and possibilities of specific modules, but accountants are occupied with one specific question inside Financial module. They are looking for an easy but flexible mapping process of journal vouchers and documents or preciously business events. One of the simplest approaches is the mapping accounts for every existing document. That approach is the fundamental of Synesis, one of the most used Croatian ERP systems for micro firms. It is easy but not flexible way, because it does not allow adding specific accounting entry for existing documents or creating custom document and mapping to custom posting.

Accounting customization is maybe sufficient for everyday purchase and sales events, but flexibility is needed in a production. Production accounting is one of the most complex areas that hardly could be the same even in two organizations from the same branch. Raw materials, waste, semi manufactured products and products circulate every day. Products could be return back to production as a raw material, or can change place of warehousing. Occasionally, firm can sell the part of waste. Every mentioned event is followed by specific document (production orders, warehouse movement, receipts, shipments etc.) and posted to general ledger accounts. Analytics of accounts are changed at the same time, so the insight into the financial and natural state should be available to stockmen, commercial travelers, financial officers or every department with the appropriate approval.

From the accountant perspective, document based posting setup in modern ERP applications is not so customizable or similar to the journal vouchers that are generated by document based accounting. That is influenced by the integrity of modules on the one side and by database design on the other side. Database designer aim is to create transparent database without redundancy. At the same time, business processes cause redundancy in all ERP modules. That requests a lot of complex queries to the different database tables that are connected with the One-to-One, One-to-Many or Many-to-Many relationships realized by foreign key fields. Number of tables depends on business processes, but also on a database designer. Will he create a single table with a lot of fields or more interconnected tables, for some specific module?

Insight in the Postgresql database of Openbravo ERP

system reveals 476 tables.

Complexity of ERP systems for large companies is even bigger. SAP ERP database count 14.000 tables and 650 fields for material items [11].

V. MODEL-VIEW-CONTROLLER (MVC)

Table II identifies the software design concepts that have proved themselves in modern ERP system development. It recognizes 3-tier architecture in 7 of 10 ERP applications and 5 of 10 have MVC pattern.

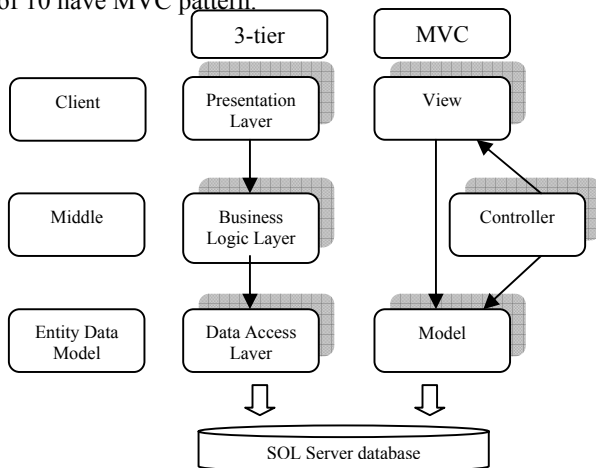


Fig. 1 Comparison of software design pattern

MVC is one of the most popular design patterns in 3-tier applications. Difference between 3-tier and MVC 3-tier is in a communication direction. While non-MVC 3-tier applications have a bi-directional communication that always passes through the Middle tier, Fig. 1 shows unidirectional communication in MVC framework. *Controller* has a complete access to both the *View* and the *Model*. *View* has a limited access to the *Controller* because the *Controller* can be replaced at any moment. The *Model* sends notification to the *View* when its data has been modified in order to redraw its content by the *View*, but *Model* does not care for inner workings of the *View*. MVC provides a mechanism for binding HTML form fields to a business model object through an *UpdateModel* command in the controller layer. When the form is submitted to the controller, the controller will bind and map the values of the form to the business object.

Although MVC model was defined in the 70s (invented by Smalltalk programmer Trygve Reenskaug), its popularity arose with the Ruby on Rails web application framework. Microsoft has also recognized its advantages and released the first version of ASP.NET MVC framework in 2009.

VI. OBJECT ROLE MODELING

In 2-tier applications *client* directly communicate with the *database*. Access to relational database is based on queries according to identification fields. Complex queries for report generating are quite large challenge for programmers and the system performance. By adding Data Layer, applications become 3-tiered.

3-tier data-centric application development usually starts with the designing of a database and the Data Access Layer, or

a *Model*. Software development has a progress in that field. Modern ERP applications use Object Role Modeling (ORM) for creating Data layer in 3-tier applications.

ORM is a powerful method for designing and querying database models at the conceptual level, where the application is described in terms easily understood by non-technical users. In practice, ORM data models often capture more business rules, and are easier to validate and evolve than data models in other approaches [12]. Developer works with object models instead of complex queries based on primary keys. Entity framework runtime will handle database connections, commands generation, query execution, object materialization and the details of persisting changes back to the database [13]

ADO.NET Entity Framework is one of the object-relational mapping (ORM) frameworks for .NET. It was released in July 2008 as part of the Visual Studio SP1. The core benefit is lack of concern for database structure because it translates tabular data of relational database into objects through the Entity Data Model (EDM).

EDM is an xml client-side model with permission to restructure the database tables, views and relationships according to a business domain regardless the normalized schema that is design by database administrator [13]. The fact that *Model* is in a memory speeds up access to a data in opposite to database querying. Disadvantage of this approach is a loss of data in a case of unpredictable crashes.

Entity framework also allows generating database script from data model. That option, together with the independency of entity syntax, allows creating database from the scratch, or inside other SQL server, without worry about database differences.

VII. ACCOUNTING INFORMATION SYSTEM AS ENTITY DATA MODEL

General Ledger (G/L) is the base module of Accounting Information System. Its structure is arranged by Croatian Accounting Act, Croatian and International Financial Reporting Standards. Its purpose is evidence of all business events according to double-entry principles. G/L balance sheet contains synthetic accounts of assets, equity, liabilities. Inside liabilities there is *financial result* account, obtained by subtracting income and expense accounts. Off balance sheet items are divided into assets and liabilities accounts. G/L fundamental financial reports (balance sheet, income statement, cash flow statement, equity statement) are also structured according to the legislation. G/L also has *other reports* for Tax Administration or internal usage [5].

Analytical bookkeeping represents value and natural details about specific G/L accounts. Every accounting system has different number of analytical evidences, but the usual ones are assets, payable and receivable analytics (advance, payrolls and loans), analytics of raw materials, production and merchandise accounts.

Subsidiary ledgers are: cash book, invoice books, share books etc. They mostly derived from analytical evidence.

Reference [5] recognizes two basic groups of accounting applications, from the perspective of analytic and synthetic connection. First group of systems contains analytic accounts

inside synthetic evidence and the second separates G/L accounts from analytics module.

AIS looks different from the *entity data model* perspective. One of the possible design solutions can be seen on Fig. 2. General Ledger is represented with the Journal table, although they are not synonyms by accounting syntax. Basic control of double entry accounting is equality of summation of Debit and Credit fields.

Possible types in Chart of Account are asset, equity, liability, income or expense. Relationships between entities are One-to-Many. Analytical bookkeeping is represented with one entity, but it is easy to add new, according to system demands.

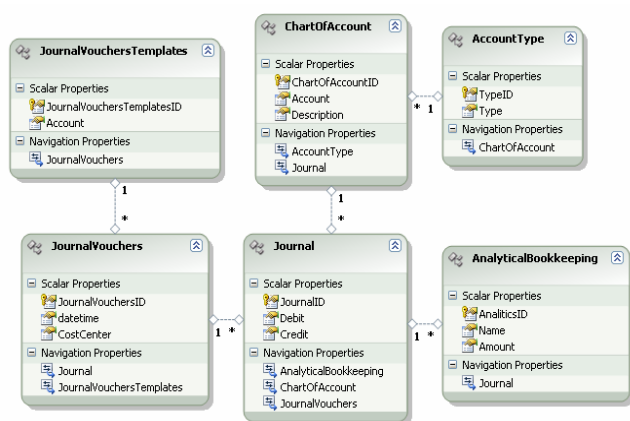


Fig. 2 Main part of AIS as entity data model

To get Journal with all needed data (date and time, cost center, accounts and description of account), *Controller* simple accesses to *Model* as on Fig. 3. *View* is html file that reads *Journal* items.

```

1 using System;
2 using System.Collections.Generic;
3 using System.Linq;
4 using System.Web;
5 using System.Web.Mvc;
6 using MvcErp.Models;
7
8 namespace MvcErp.Controllers
9 {
10     [HandleError]
11     public class HomeController : Controller
12     {
13         MvcERPEntities entities = new MvcERPEntities();
14
15         public ActionResult Index()
16         {
17             var journal = entities.Journal
18                 .Include("ChartOfAccount")
19                 .ToList();
20             return View(journal);
21         }
22     }
23 }

```

Fig. 3 Controller access to entity data model

VIII. CONCLUSION

Adoption of Openbravo has increased in developing countries. There are nearly 5,000 deployments across 19 Asia-Pacific countries, representing a growth rate of 138 percent in the region from 2010 [14]. Despite the fact that open source business application is a new concept, complex nature of open source ERP development requires proven technology. Openbravo and similar projects use MVC pattern.

Rise of Croatian Entrepreneurship in the nineties of the last century encouraged development of the small Microsoft DOS applications written in Clipper programming language and dBase database management system. Applications represented independent modules of accounting information systems.

These modules were not as integrated or automated as today. They did not have today capabilities like planning or "what if" analysis. Accountant department had a monopoly on accounting information, but the use of software was easy, without the need for lots of hours of training courses.

Access to accounting information today is a matter of assigned roles and permissions of administrator. It is not a technical challenge any more. Sophistication brought up accounting information to any department. Negative issue of sophistication is accounting setup complexity as a result of modules interaction. Complexity increases educational and support costs.

Croatia has a tradition in developing ERP applications for small and medium enterprises that make 98% of Croatian Entrepreneurship. It also has cheaper developers than EU countries. Stronger part of Croatian ERPs is their Financial & Accounting module as the most important and as one of the first business functions to be computerized. But still, Croatia is not recognized as an international player in developing simple, highly customizable and robust AIS framework, although reducing ERP complexity, time and costs of implementation and maintenance, is one of the most recognized problems of business today. Efforts in that way may attract venture capital, as Spanish Openbravo project did, and create opportunity to compete on an international market.

Business software technology is easy accessible resource. Croatia has knowledge, only misses the strategy.

REFERENCES

- 1] http://www.unit4.be/files/The_High_Cost_of_Change_of_ERP.PDF (Access: (Feb 1 2011)).
- 2] <http://openacct.wordpress.com/2011/02/28/openbravo-launches-open-source-erp-for-smb/> (Access: April 22 2011)
- 3] <http://www.opensource.org/licenses/ms-pl.html> (Access: April 5 2011).
- 4] G. Glenn, *Enterprise Resource Planning 100 Success Secrets*, Emereo Publishing, Newstead, 2008, pp. 14, 21.
- 5] I. Mamić Sačer, K. Žager, *Računovodstveni informacijski sustavi*, Hrvatska Zajednica Računovoda i finansijskih djelatnika, Zagreb, 2007, pp. 145-232
- 6] Marston, S. R. et al., (2009) Cloud computing: The Business Perspective Available on: http://papers.ssrn.com/sol3/papers.cfm?abstract_id=1413545 (Access: Jan 4 2011)
- 7] Antonic, J. (2009) A Brief History of Cloud Computing Available on: <http://www.undertheradarblog.com/blog/a-brief-history-of-cloud-computing/> (Access: Jan 4 2011)
- 8] Buyya, R. et al., Market-oriented cloud computing: vision, hype, and reality or delivering it services as computing utilities. 10th IEEE International Conference on High Performance Computing and Communications September 2008, Dalian, China. Available on: http://www.buyya.com/papers/hpcc2008_keynote_cloudcomputing.pdf (Access: Jan 19 2011)
- 9] <http://www.openbravo.com/product/erp/technology/> (Access: March 03 2011)
- 10] http://wiki.openbravo.com/wiki/Functional_Documentation/Financial_Management/Accounting (Access: March 03 2011)
- 11] K. Fertilj et al. *Komparativna analiza programske potpore informacijskim sustavima u Hrvatskoj*, Sveučilište u Zagrebu, FER, Zagreb, 2002. Available on: <http://www.zpr.fer.hr/zpr/LinkClick.aspx?fileticket=iCZzjAPMIS4%3d&tabid=104&mid=457&language=hr-HR> (Access: (March 11 2011)
- 12] <http://www.orm.net/> (Access: (April 26 2011)
- 13] J. Lerman, *Programming Entity Framework*, Second Edition, O'Reilly Media, Inc., Sebastopol, 2010, pp. 3, 20
- 14] <http://www.openbravo.com/about-us/press-room/news/162/openbravo-erp-adoption-in-asia-pacific-surges-138-percent-in-2010.php> (Access: