

Comparative Study of Open ERP and its Technologies

Navita Jindal
Faculty of CSE Department
RIMT-IET, Mandi Gobindgarh
Punjab,INDIA

Kanwalvir Singh Dhindsa
Faculty of CSE & I.T Department
BBSBEC, Fatehgarh Sahib
Punjab, INDIA

ABSTRACT

ERP software can be extremely complex, as it must to meet the need and expectations of some of the largest and most sophisticated companies in the world. OpenERP is a modern enterprise management software based on Open Object, a modular, scalable and intuitive Rapid Application Development frame-work written in python. This review paper focuses OpenERP as Open source alternative to Enterprise level implementations for small to mid scale Enterprises (SMEs) and how open source ERP development can affect these enterprises. The paper will discuss core technologies and methodologies that are necessary for low cost ERP implementations in SMEs.

Keywords

OpenSource ERP Development; OpenERP; SAPAG; ORM; Python.

1. INTRODUCTION

Enterprise Resource Planning (ERP) software comes as a suite of utilities that divides business processes broadly into the following conceptual areas in order to make the structure of very complex software manageable, manufacturing, supply chain, financial, project management, human resource management and customer supplier relationship management. In other words, there's something for everyone who is involved with the company's management. As Enterprises mature and their requirements grow more and more functionalities are added to the system which results in increased complexity of the ERP software and implementation [9]. Opensource ERP is accountable for the organizations seeking continuous adoption of the software to changing processes and needs. OpenERP released under the AGPL license features Project Management, HR, CRM, Sales, Accounting, Manufacturing, and Inventory. OpenERP uses a three-tier architecture written in Python [6]. The application tier is written as a core and multiple additional modules that can be installed or not to create a particular configuration of OpenERP. The functionality of a module is exposed via XML-RPCs. Modules can insert data in the database during installation by providing XML or CSV or files.

2. OPENERP ARCHITECTURE

Some large and medium-large companies implement OpenERP with their own internal resources. They prefer to have their own IT service in charge of maintenance. Such companies can do the implementation work themselves internally. Figure 1: shows the basic architecture of OpenERP.

An Open ERP system is formed from three main components such as:

1. Database Layer (PostgreSQL)

Open ERP uses PostgreSQL as the default database for all its functionality [11]. The PostgreSQL database server contains all of the databases that contain all data and most elements of the Open ERP system configuration.

2. Open ERP application server (Middle Layer)

The Open ERP application server, which contains all of the enterprise logic and ensures that Open ERP runs optimally. The Server itself is written in Python language. Open ERP application server is released under Aeffro GPL License [3].

3. Client Layer

The web server, a separate application called the Open Object client-web, which enables connecting to Open ERP from standard web browsers and is not needed when system can connect using a GTK client [7]. The client-web component can be thought of as a server or a client depending on the user's viewpoint. It acts as a web server to an end user connecting from a web browser, but it also acts as a client to the Open ERP application server just as a GTK application client does.

3. OPENSOURCE ERP BENEFITS

Both Open Source ERPs and owners involve complex implementation processes in which companies and software are not familiar to changes in their processes. This mutual adaptation generally involves consultancy companies so that processes are less traumatic and less costly in what regards time and money. Benefits when choosing Open Source Systems are described as follow:

1. **Better adaptability:** Due to the availability of the software source code and its free manipulation, the customization tends to be easier [7]. Thus, the need to customize it according to local laws and the company peculiarities, among others, are always necessary, independent of the area of work.
2. **Minimum supplier dependency:** Once a private solution is achieved, the company will be a hostage/dependent of the company which owns the software/project [6]. Therefore, in case the owner company leaves the market or project, the continuity of the maintenance and updation of the ERP on the client company may be seriously jeopardized, since it will not have access to the source code of the software.
3. **Cost reduction:** The Open Source ERP has no costs on license acquisition and usually does not need expensive equipment to be nicely performed [4].

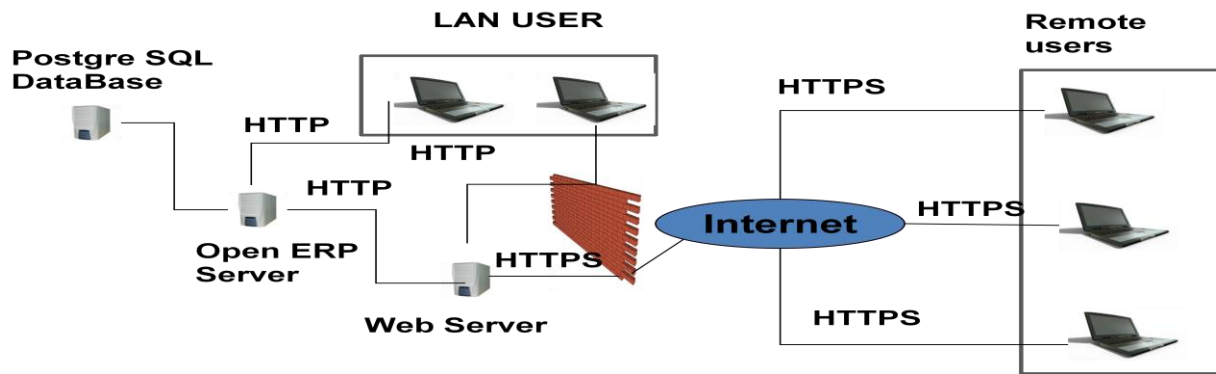


Fig. 1. The architecture of OpenERP [3]

4. **Easier integration:** ERP solutions touch a company's each and every aspect from warehousing to accounting. ERP solution should easily integrate with company's existing IT infrastructure components, such as directory services, application servers and storage arrays. With standards-based interfaces, Open source solutions are compatible with multiple technologies, including support for databases, lowest-cost commodity, hardware, operating systems and utilities [15].
5. **Quality:** Open Source ERPs are distinguished by a superior quality on the technical level. These superior quality standards can be due to the commitment of vendors (Open Source ERP vendors) against technical challenges [6].
6. **Vendor independence:** By adopting an Open Source ERP, the client is not at the mercy of the vendor or product owner. The community provides the support of the product and the client is in a powerful position to exert a proactive pressure over the vendor.
7. **Freedom for upgrade:** An organization that chose to adopt an Open Source ERP can choose to upgrade it whenever they want. They can choose to do the upgrade themselves or at the best price by sending several requests for proposal to several competing third parties and choosing the best offer at the end.
8. **No hidden costs:** Many proprietary ERP vendors attract clients by proposing packages at acceptable prices but which soon turn out to be limited, namely by the number of users [7]. Once an organization adopts a licensed ERP package or suit, it finds itself unwillingly locked in a costly battle to scale up the system if the need arises to increase the number of the ERP users. Possibility of try to keep control over the ensemble of functional modules that a client's organization may need.

4. OPENERP TECHNOLOGIES

OpenERP is available at low price compared to other form of ERP services and the industry buying services do not require to spend on customization, integration, implementation of application etc. The technologies of Open ERP are described as below:

4.1 Python

The Python programming language is establishing itself as one of the most popular languages for scientific computing. Python is a general purpose interpreted, interactive, object-oriented and high-level programming language [1]. In the late eighties and early nineties Python was created by Guido van Rossum the National Research Institute for Mathematics and Computer Science in the Netherlands. Its design was inspired by a number of other languages, including C, Modula-3, C, C++ and particularly the educational language ABC. It is similar like Perl as Python source code is also now available under the GNU General Public License (GPL). Like Perl, Python is run as a large and successful open source project.

4.1.1 Python Features

Python features are described as below:

- **Easy to learn:** Python structure is very simple. It has clearly defined syntax and relatively few keywords.
- **Easy to read:** Python code is very easy much more clearly defined and visible to the eyes.
- **Easy to maintain:** Python source code is very easy to maintain [2].
- **Portable:** Python is easily run on a various hardware platforms and has the same interface on all platforms.
- **Extendable:** Python is easily extendable as it adds low-level modules to the Python interpreter which allow the programmers to add to or customize their tools to be more efficient.
- **GUI Programming:** GUI applications are supported which are created and ported to many system calls, libraries, and windows systems.
- **Scalable:** Python provides a better structure and support for large programs than shell scripting.

4.2 PostgreSQL

It is an object relational database management system. It is freely available. PostgreSQL was developed by university of California and uses the simple BSD license. It is very advanced database system in Open source area. It support most parts of SQL 2003 standards and has variety of its own extension. It also offers definition of rule and trigger to control access to database object. PostgreSQL community is also known for its rapid response to user question and issues [10]. The PostgreSQL global development group is famous for its fast turnaround for its bugs which make PostgreSQL is stable and secure database system.

4.2.1 Reason for using PostgreSQL

- Proprietary database are expensive with complex pricing structure: As it is free of cost it provide a viable alternative without sacrificing significant performance or feature for a low TCO to highly adaptable solution.
- Scale as the business grows: It is highly scalable database known to scale up to 6TB of data with no extra license cost to move up to bigger server [11].
- It delivers the reliability and security so need not to worry about losing the data.

4.2.2 PostgreSQL Features

The features of PostgreSQL are described as below:

- Stored procedure which may be written in 12 different programming languages.
- Has unique database extensibility for domain specific solution like genetic and cryptography [12].
- Drivers available for all major programming language.

4.3 Open Object

Open Object is a smart open source professional rapid application development framework in python. To build up the best enterprise management software, a perfect organization between all Open ERP's actors is required. So that organization can benefit and leverage the contributions and feedbacks from the community, the market knowledge and creation from partners and the quality control. The Enterprise must focus on creating a fully Open Source development methodology [8]. Object-relational mapping in computer software is a programming technique for converting data between incompatible type systems in object-oriented programming languages. This creates, in effect, a "virtual object database" that can be used from within the programming language. As some programmers opt to create their own ORM tools there are some free and commercial packages available that perform object relational mapping. OpenERP has an Object-Relational Mapping technology on top of the database that allows to directly manipulate objects instead of the database queries [14].

5. COMPARISON: OPENERP TECHNOLOGIES VS SAP AG TECHNOLOGIES

OpenERP is an open source business application aimed at leveraging the power of open source community to develop a dominant ERP system. ERP implementations have been simplified by OpenERP across industries by making available basic functionalities.

SAP AG is a German based multinational software corporation which makes enterprise software to manage business operations and customer relations. Its Headquarter is located in Walldorf, Baden-Württemberg, Germany, with regional offices around the world. Many large and mid size company use SAP which is the leader in development software and software-related service.

1. Programming layer: Python Vs ABAP

Python is a general-purpose, high-level programming language which emphasizes code readability. Python's has remarkably simple and elegant syntax that allows programmers to express concepts in fewer lines of code which would be possible in languages such as C, and the language provides constructs intended to enable clear programs on both a small and large scale[1].

Whereas SAPs ABAP stands for (Advanced Business Application Programming, originally Alleghenies Berichts Aufbereitungs Prozessor, German for "general report creation processor) is a high-level programming language created by the German software company SAP [13]. It is currently positioned, alongside the more recently introduced Java, as the language for programming the SAP Application Server, part of its NetWeaver platform for building business applications. The syntax of ABAP is similar to COBOL.ABAP is especially been developed for use in report Generation and can only be used in the context of SAP netweaver [15], Whereas Python works very well on various architectures and platforms. Another Advantage of Python over ABAP is its modern Object oriented nature makes it possible to implement features such as OpenERP's ORM. Python is also Open source which further makes it easily available and it can be used anywhere without any enterprises consent. Table 1 summarizes the characteristic of python Vs ABAP.

Table 1: Python and ABAP characteristic comparison

S.No.	Characteristics	Python	ABAP
1	Language Design	General Purpose	SAP Specific
2	Style	Simple and Elegant	Complex and business specific
3	Syntax	Modern Object Oriented	General Report creation processor
4	Platform Independence	YES	No (SAP Specific)
5	License	Open Source	Proprietary
6	Version	2.7.3	4.1
7	Development Time	Less	Very Large
8	Debug Capabilities	Diverse	SAP Specific

2. Database layer: PostgreSQL Vs Oracle

Oracle is much more scalar in performing as compared to PostgreSQL because of exhaustive research and development backed by huge International Corporation. Oracle is free of cost for personal and education purposes but Oracle charge per CPU

for enterprise uses. PostgreSQL benefits from its open source nature and can be used in any operating system [10]. The PostgreSQL is based on BSD (Berkeley Software Distribution) whereas Oracle database is proprietary software.

In case of ERP system PostgreSQL can be much suitable because of cost free nature and easily available for SMEs. Table 2 summarizes the characteristic of PostgreSQL Vs Oracle.

Table 2: PostgreSQL and Oracle characteristic comparison

S.No.	Characteristic	PostgreSQL	Oracle
1	Database Design	Less Scalable	Highly Scalable
2	Style	Closely Matches Oracles	Oracle Industry Standard
3	SQL Standard	PostgreSQL	PL/SQL
4	Platform Independence	YES	NO
5	License	Free and Open Sourced	Proprietary
6	Version	8.7.2	10.0.2

3. Presentation layer: Modern web interface Vs classical desktop

The two most important differences between the two interfaces are:

The web interface has fewer interactive notification options but runs on any operating system which is able to run either Internet Explorer 6.0 or Firefox without installing any client software

Whereas the any GUI based product can only be used on operating systems which provide GUI libraries and offers additional notification options. Because both interfaces can be used simultaneously on an O.S for example the OpenERP's GTK Client and the web interface can easily be accessed from the same machine. There are some good reasons to use both of them. If the interface of the application can be implemented as a web service or application the enterprises should seriously consider it as an option.

The biggest benefit of a Web interface is that it can run in the browser and is its independent from client OS. Today most computer users have more than one computer which usually connected in a network. The distribution of different operating systems is getting larger every day so to get maximum portability of the user interface. The application can be started once on one computer and accessed from anywhere. Table 3 summarizes the characteristic of Web Vs Classical Desktop.

Table 3: Web and Classical Desktop characteristic comparison

S.No.	Characteristic	Web	Classical Desktop
1	Notification Support	Smaller Support	Larger support
2	Platform Independence	YES	No (OS API Specific)
3	License	Mostly Free	Depending upon OS
4	Development Time	Lesser	Very Large
5	Interface Portability	Can be Made Completely Portable	Depends upon Cross Platform API Like GTK.
6	Accessibility	Can be Accessed from Anywhere	OS Dependent
7	Security and Control	Much more Secure	Less Secure

4. Programming methodology: OpenERP's ORM Vs SAP

ORMs have many good features. They can handle much of the repetitive work of copying database columns to object fields. They can handle converting the language's which is inherent to the appropriate database type. They handle one-to-many relationships very easily as well by instantiating nested objects [8]. Keeping ORM in mind the database is designed with the strengths and weaknesses which save a lot of work in getting data in and out of the database. But it is needed to know how it handles polymorphism and many-to-many relationships if they are need to be mapped. It's these two domains that provide most of the 'impedance mismatch' that makes them difficult to implement.

For applications that are transactional, i.e. it needs to make a request, get some objects, traverse them to get some data and render it on a Web page, the performance lag is small, and in many cases ORM can be faster because it will cache objects as it is seen before otherwise it would have queried the database multiple times. For applications that deal with large number of database rows per request, the ORM lag is much heavier, and the caching that ORM modules requires big and memory hogging burden, which is usually the case in Large Enterprises.

For any large-scale application both approaches are necessary. The ORM is best suited for SMEs as it reduces the implementation and maintenance cost considerably. Table 4 summarizes the characteristic of ORM Vs SAP.

Table 4: ORM and SAP characteristic comparison

S.No.	Characteristic	ORM	Classical(SAP)
1	Features	Data abstraction and manipulation	Few and hard to implement
2	Database Independence	YES	NO
3	Performance	Scalable or SME's but not suitable for large Enterprises	Faster than ORM
4	Rapid Application Development	YES	Not possible
5	Programming layer	Existing	Non Existing
6	Inheritance	Supported	Not Supported

4. CONCLUSION

Today, it's the era of online apps and services they help easily connecting more and more people throughout the world effortlessly. The new generation in which they live in today already apprehends a totally different way the world of information, the tasks execution and the messaging. The world of tomorrow will be dematerialized. The personal computer that is being used and pioneered exclusively will be used only by some specialists to maintain operational dematerialized environments and the ERP will no longer exist as a monolithic structure. The many support activities will be automatically initiated and controlled by neuronal systems learning from their mistakes and sharing their experiences with the other computers in a viral and community way. The activities with no added value which consume time and money will disappear. Companies will then be able to concentrate on their main activities i.e. production and services, sales and Research and Development.

OpenERP is truly a modern Enterprise Management Software based on scalable architecture. OpenERP used Python as the default language whereas SAP used ABAP whose syntax is similar to COBOL which is mainly used for report generation. The main advantage of python over ABAP is object oriented nature make it possible to implement feature such as Open ERP ORM. PostgreSQL benefits from its open source nature and can be used in any O.S. but oracle charge per CPU for enterprise level. In Open ERP it is benefit that both the client and web interface can be easily accessed from the same machine at same time. Open ERP has object relational mapping technology on top of database that allow to directly manipulate object instead of database queries. In this paper different technology of

OpenERP and SAP are studied and conclude that Open ERP use modern technology to tackle the needs of a modern Small to Middle scale enterprises.

However from a functional point of view, OpenERP must extend its industry solution offer by integrating the specific requirements of some specific sectors. OpenERP must also keep on developing new functionalities in which it lacks. These functionalities are expected by any major clients for instance: Advanced Treasury Management, Budgetary Management for Public Sector, etc.

REFERENCES

- [1] Pedregosa, F., Varoquaux, G., Gramfort, A., Michel, V., Thirion, B., Grisel, O., & Duchesnay, É. 2011. "Scikit-learn: Machine learning in Python", *The Journal of Machine Learning Research*, Vol. 12, pp. 2825-2830.
- [2] Sanner, M.F.1999. "Python: a programming language for software integration and development", *Journal Mol Graph Model*, Vol. 17, No.1, pp. 57-61.
- [3] Al-Saleem, S. M. 2013. "A Comparative Analysis and Evaluation of Open Source ERP Systems", *International Journal of Computer Science and Network Security (IJCSNS)*, Vol. 13, No. 4, pp. 24-28.
- [4] Dinesh, E., Vetrivel, T., 2013. "Benefits of Open Source ERP for SMEs in India", *Global Journal*, Vol. 2, Issue No. 2, pp.157-158.
- [5] Gripe, F.G.dos S., Rodello, I.A., 2011. "A theoretical analysis of key points when choosing Open Source ERP systems", *Journal of Information Systems and Technology Management*, Vol. 8, No. 2, pp. 441-458.
- [6] Johansson, B., & Sudzina, F. 2008. "ERP systems and open source: an initial review and some implications for SMEs", *Journal of Enterprise Information Management*, Vol. 21, No. 6, pp. 649-658.
- [7] Herzog, T. 2006. "A Comparison of Open Source ERP Systems", Institute of Software Technology and Interactive Systems, Business Informatics Group, Vienna University of Technology.
- [8] Keith, M., & Schnicariol, M. 2010. "Object-Relational Mapping", *In Pro JPA*, Vol. 2, pp. 69-106. Apress.
- [9] Umble, E.J., Haft, R.R., and Michael, M., 2003. "Enterprise resource planning: Implementation procedures and critical success factors", *European Journal of Operational Research*, Vol. 146, Issue 2, pp. 241-257.
- [10] Levshin, D. V., & Markov, A. S. 2009. "Algorithms for integrating PostgreSQL with the semantic web", *Programming and Computer Software*, Vol. 35, No. 3, pp. 136-144.
- [11] Lo, E., Binnig, C., Kossmann, D., Tamer Ozsu, M., & Hon, W. K. 2010. "A framework for testing DBMS features", *The VLDB Journal—The International Journal on Very Large Data Bases*, Vol. 19, No. 2, pp. 203-230.
- [12] Stonebraker, M., & Kemnitz, G. 1991. "The POSTGRES next generation database management system", *Communications of the ACM*, Vol. 34, No. 10, pp. 78-92.

- [13] Weinhardt, C., Anandasivam, A., Blau, B., & Stober, J. 2009. "Business models in the service world. IT professional", Vol. 11, No. 2, pp. 28-33.
- [14] Gacek, C & Arief, B., 2004. "The Many Meanings of Open Source", *IEEE Software*, Vol. 21, No. 1, pp. 34-40.
- [15] McCarthy, B., & Hawking, P. 2002. "Teaching SAP's ABAP programming language to IS students: Adopting and adapting web-based technologies", In the Proceedings of Informing Science and IT Education Conference, Cork, Ireland, pp. 995-1000.