Bogdan MICU GHILIC
George STANCU

ADAPTING OPEN-SOURCE SOFTWARE FOR ENTERPRISE

Keywords
Open-source
Software enterprise
CRM
ERP
OFBIZ

JEL classification
M15

Abstract

Low developed companies do not use ERP solutions for resource management, they don't have CRM systems for tracking customers, online presence is low and consequently they are selling low volumes. Their work does not generate enough capital to buy specialized software and the local community is too small to create open-source software for them.

When adapting an open-source enterprise solution, from a certain level of integration, it is necessary to align to the Romanian legislation. For this you need more than a simple translation. For example, in accounting, the Romanian chart of accounts should be used, the local fiscal calendar, the rules of the Romanian legislation and the specific identifiers for Romanian companies.

The solution:

Open source applications can be localized. A small community cannot create complex system starting from nothing, but they can adapt existing ones already created and used all over the world.

Given the individual characteristics of the Romanian market, open source ERP, CRM and e-commerce will be integrated by specialized consultancy agency in Romania just as is done in France (Compiere), Spain (Openbravo), Belgium (OpenERP) or the United States (Apache oFBiz).

This will not generate license costs for the companies, only the expenses for hardware and implementation consultancy. If we can create a common infrastructure, hardware cost will be waived for cloud SaaS offering, in opposition with the on-premise option.

OPEN SOURCE TRENDS

Low developed companies do not use ERP solutions for resource management, they don't have CRM systems for tracking customers, online presence is low and consequently they are selling low volumes. Their work does not generate enough capital to buy specialized software and the local community is too small to create open-source software for them.

When adapting an open-source enterprise solution, from a certain level of integration, it is necessary to align to the Romanian legislation. For this you need more than a simple translation. For example, in accounting, the Romanian chart of accounts should be used, the local fiscal calendar, the rules of the Romanian legislation and the specific identifiers for Romanian companies.

The solution:

Open source applications can be localized. A small community cannot create complex system starting from nothing, but they can adapt existing ones already created and used all over the world.

Given the individual characteristics of the Romanian market, open source ERP, CRM and e-commerce will be integrated by specialized consultancy agency in Romania just as is done in France (Compiere), Spain (Openbravo), Belgium (OpenERP) or the United States (Apache oFBiz).

This will not generate license costs for the companies, only the expenses for hardware and implementation consultancy. If we can create a common infrastructure, hardware cost will be waived for cloud SaaS offering, in opposition with the on-premise option.

Choosing the right enterprise open-source solution

Enterprise open-source IT platforms

ERP Package	Language Base	License	Other Info	Developer Country
A1.iO	Java	ATOL	ERP for Public Sector, Campus Management, Healthcare, Logistics A1.iO	Worldwide
Adaxa Suite	Java	GPL	Integrated ERP built on Adempiere	Australia/New Zealand
Adempiere	Java	GPL	Started as a fork of Compiere	Spain
Compiere	Java	GPL/Commercial	Acquired by Consona Corporation in June 2010	US
Dolibarr	PHP, MySQL	GPL		
EpesiBIM	PHP, MySQL	MIT license	Web based application	Poland, USA
ERP5	Python, Zope, MySQL	GPL	Based on unified model	Brazil, France, Germany, Japan Sénégal
ERPNEXT	Python, JavaScript, MySQL	GPL	ERP for small and medium businesses	India
Fedena	Ruby, MySQL	Apache License	ERP for Schools/Universities	India
FrontAccounting	PHP, MySQL	GPLv3	Web-Based system	
GNU Enterprise	Python	GPLv3		
HeliumV	Java	AGPL	ERP for small and	Austria,

ERP Package	Language Base	License	Other Info	Developer Country
			medium businesses	Germany
JFire	Java	LGPL		
Kuali Foundation	Java	ECL	for higher education, by higher education	
LedgerSMB	Perl, PostgreSQL	GPL	started as a fork of SQL-Ledger in 2006	Worldwide
OFBiz	Apache, Java	Apache License 2.0	ERP for small and medium businesses	
Openbravo	Java	Openbravo Public License (OBPL), a free software license based on the Mozilla Public License (MPL)		Spain
OpenERP	Python, PostgreSQL	AGPLv3	OpenERP version 7.0 was released on 12/21/12, OpenERP was formerly known as Tiny ERP	Belgium, India, USA
Phreedom	PHP, Javascript, MySQL	GPLv3	Expanded from Phreebooks accounting engine	USA
Postbooks	C++, JavaScript, PostgreSQL	CPAL	Produced by XTuple, uses Qt framework	
SQL-Ledger	Perl, PostgreSQL	GPL		
Tryton	Python	GPLv3	Started as a fork of OpenERP	
WebERP	PHP, MySQL	GPLv2	LAMP based system	

Source: Wikipedia 01.06.2013 http://en.wikipedia.org/wiki/List_of_ERP_software_packages

APACHE OPEN FOR BUSINESS PROJECT

About Apache Open for Business project

Open for Business (OFBiz) is a suite of enterprise applications built on a common architecture using common data, logic and process components. The loosely coupled nature of the applications makes these components easy to understand, extend and customize.

The tools and architecture of OFBiz make it easy to efficiently develop and maintain enterprise applications. This makes it possible for the creators and maintainers of the project to quickly release new functionality and maintain existing functionality without extensive effort. It also makes it easy to customize and extend existing functionality when there is a specific need.

The architecture alone makes it easier to customize the applications for a particular need, but many of the best flexibility points in the system would be meaningless and even impossible if the system was not distributed as open source software. OFBiz is licensed under the Apache License Version 2.0 which grants anyone the right to customize, extend, modify, repackage, resell, and many other potential uses of the system.

No restrictions are placed on these activities because they are necessary for effective use of this type of software. Unlike other open source licenses, such as the GPL, your changes do not have to be released as open source. There are obvious benefits to contributing certain improvements, fixes and additions back to the core project, but some changes will involve proprietary or confidential information that must not be released to the public. For this reason OFBiz uses the Apache License Version 2.0 which does not require this.

Another benefit of this open source model is that constant feedback it is received from those who are using the software. These way, there are sent countless bug fixes, improvement suggestions, and best-practice business advice from users and potential users of OFBiz. Many of the greatest features in the project were inspired by some comment or suggestion sent to the mailing lists associated with the project.

To make sure OFBiz functionality is timely and useful the development team always start by researching public standards and common usage for any component they are working on. This helps them support and use common vocabularies and gives them an instant breadth of options and features that can only be achieved through standards processes and other group efforts. It also opens doors in the future for flexible communication with other systems that are built around the same standards, both inside system and in partner or other organizations.

The applications and application components that come with the system provide a broad and flexible basis that can be used as-is with the best-practices based designs, or customized to a special needs. The applications facilitate management of everything from parties and products to accounting, customer service and internal resource and asset management.

MAJOR APPLICATION COMPONENTS

Through collaboration in a large community and an on-going development effort OFBiz hopes to include features that automate every aspect of enterprise information and knowledge. Early in the development and design of the project the creators were searching for a good initial data model to use as a foundation for the system.

They researched other ERP and CRM systems and looked at various general and system specific books. The most used book was The Data Model Resource Book, Revised Edition, Volumes 1 and 2 by Len Silverston. After a few weeks of translating the logical data models described in these books into flexible and normalized physical data models Apache they had their initial system organization. Since that time the initial data model has gone through hundreds of revisions and refinements based on real-life use of the system and compatibility with dozens of existing public standards. Because of the flexible system architecture it is easy to make changes to the data model. This is necessary for the on-going improvement of the project and makes it easy for users to make changes that suit their specific needs.

The top level applications and application logic components are organized in almost the same structure as the data model. Once the organization of one level of the system is understood moving to other levels in the system takes very little effort.

The following is a brief overview of the major functional areas of the system. Many of these have functionality that exists now and all have data elements defined that cover the needs of that part of the system. As further functionality is implemented the team continually refine the data model, but all of the major data element definitions are in place.

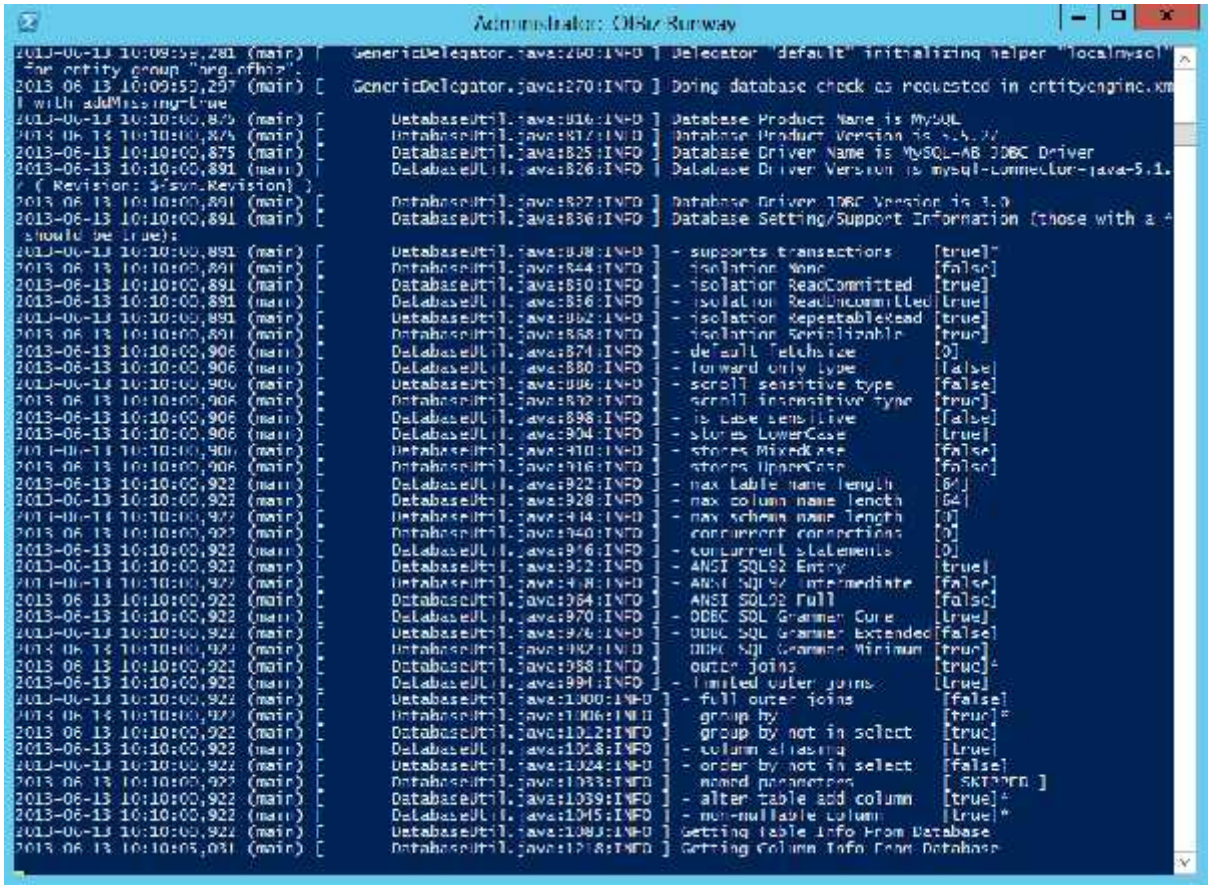

Figure 1 - OfBiz Runway

Common Data

The Common Data in the system includes entities such as Geographic Boundaries, Units of Measure, Status Codes, Enumerations, and so forth. Most of this data is seed data that is imported when the system is installed and generally needs very few changes over time. Geographic boundary and other applicable seed data is based on ISO and other standards.

Content

The Content entities are used to track data resources and structure them into general content and knowledge. They include many concepts such as: a separation of information and organization allowing a data resource to be used in many content structures; flexible organization of content including free-form association in graphs or more constrained organization in trees, lists, named maps, templates etc.; the specification of meta-data for content and data resources that can be used to implicitly organize and explicitly describe the information. Individual pieces of content can be in various text and binary formats described by standard MIME types and character encodings.

Once general maintenance tools for this information are in place, more advanced tools such as keyword based, meta-data based, and intelligent searching or text mining to automatically create additional structure or meta-data can be used to enable enterprise wide document and knowledge management.

The Content entities also include information about Web-based content such as pages and interaction with content including visits to a site (or application) and information about every request sent to the site. This is useful for tracking what users are doing with an application for security, marketing, usability and other reasons.

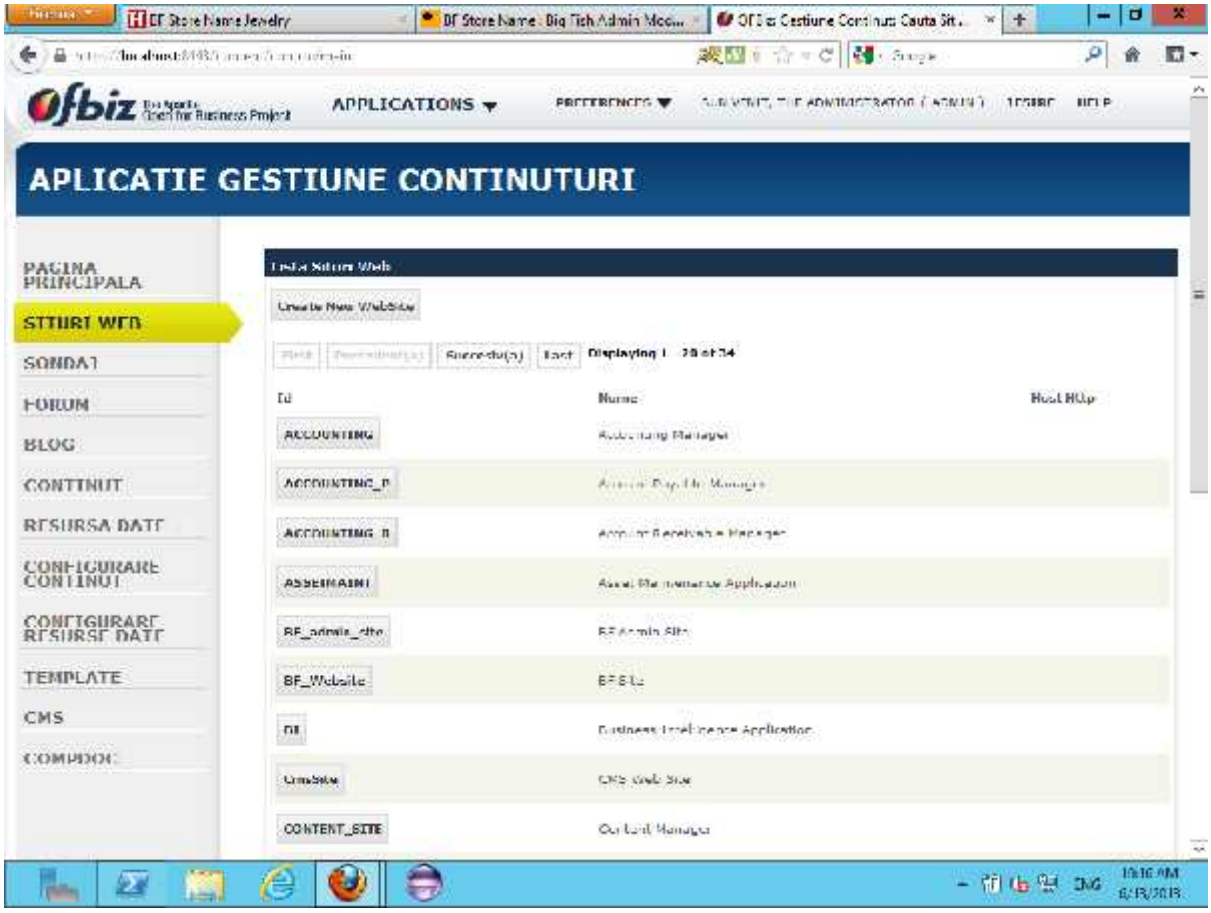

Figure 2 - OFbiz Content

Security

The Security entities are used to control access to various parts of the system and include user login accounts, login audit data, permission data, and so forth. Additional more restrictive security is accomplished through the Party entities and the association of Parties to various other data in specific Roles. For instance one user might have a Permission to view and modify all Product data where another user only has permission to view and modify the Product data if the user is associated as a Merchandiser with a category that the product is in.

Party

A Party can be either a Person, or a group of Parties. A Party Group could be a company, an organization within the company, a supplier, a customer, and so forth. Information that describes Parties or is directly related to Parties is contained in these entities.

One type of related data is Contact Mechanisms such as postal addresses, phone numbers, email addresses, internet URLs. Another is Roles that the Party acts in such as Customer, Supplier, Employee, Manager, Merchandiser, etc. Generally a single party will interact with different parts of the system in many different roles.

Another type of data that fits into the Party category is information about communication and agreements between Parties. This gets into the area of relationship management and also includes information about issues or trouble tickets that a Party may have. These entities are used along with the Work Effort entities to plan and track the research and resolution of such issues.

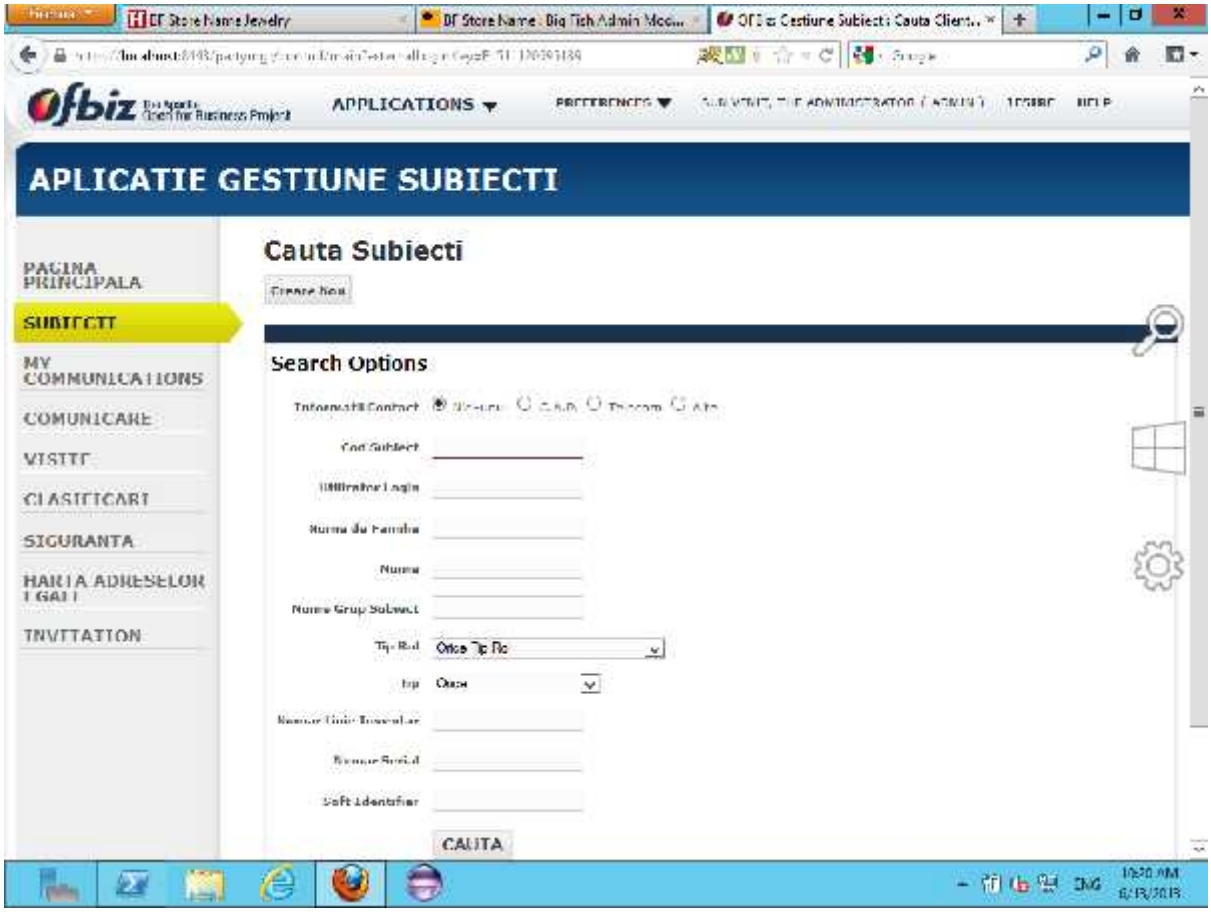

Figure 3 – Ofbiz Parties

Product

The Product entities contain information about products that are for sale or for use within a company. Products can be goods or services and there are various types of goods including raw materials, subassemblies and finished goods. Product information includes descriptive information about the products but is not used for describing actual instances (where applicable) or physical goods, that is where Inventory Items come into play. An Inventory Item contains information about where a particular good is located, the status of the item, and a serial number for serialized items or quantity on hand and available to promise amounts for non-serialized inventory.

Products can be organized into Product Categories. A single product can be a member of multiple categories and even categories themselves can be children of multiple categories and can have multiple child categories. Products can also be associated with one another to designate concepts such as variants, cross-sells, up-sells, marketing packages, etc.

Categories can be associated with different Catalogues. A Product Catalogue is essentially a starting point for all information about a particular set of products to be sold. Promotions and inventory management options are associated with each catalogue so that different sales channels can behave differently even with the same set of underlying products.

To flexibly model different types of features that products often have, there are entities for defining types of features and actual features that can be applied to products. For instance, you might say that this shirt is a size large and is a blue coloured shirt. Size and colour are types of features and large and blue are actual features applied to the specific shirt product.

Multiple prices can be associated with a single product, as can multiple costs. Different prices can be specified for different currencies, different sets of facilities (or stores), and for different date ranges.

This is a good place to introduce the wide-spread use of effective dating in OFBiz. There are two fields commonly used to express effective dates: fromDate and

thruDate. In the product price example the fromDate and thruDate are used to denote that a price goes into effect on a certain date and at a certain time and expires at a certain date and time. This can be used to keep a history of price changes, and to effectively manage temporary promotional prices.

In addition to explicitly specifying prices there are also entities and logic that uses the entities to have rules about prices. For instance you could create a rule that is in effect for a certain time period that puts all of the products in a certain category on sale. Or, you could give special prices to specific customers or groups of customers with a simple rule.

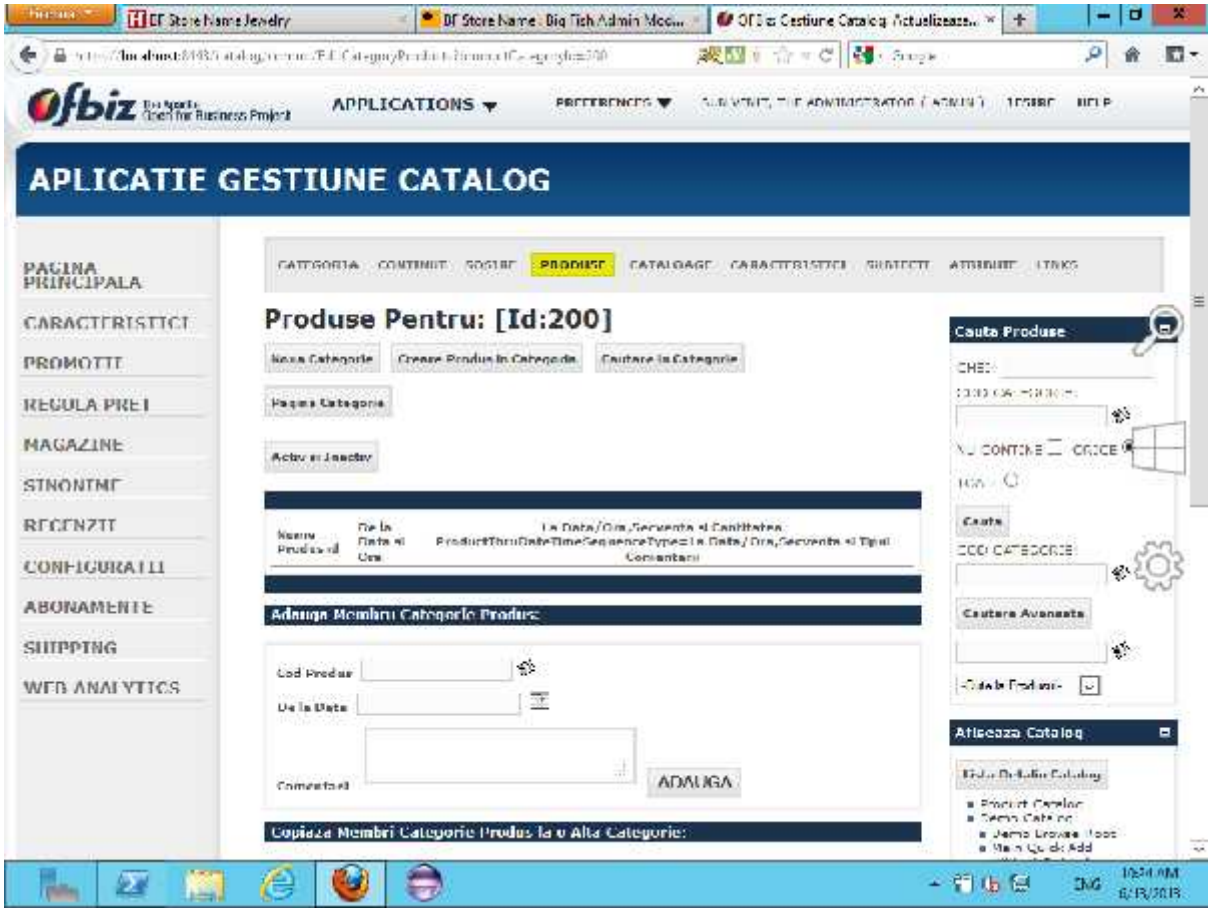

Figure 4 – Ofbiz Product

Order

The Order entities are used to manage information about sales and purchase orders and information leading up to an order. For example a request for a specific product or feature may be submitted by a customer and that would be tracked by the request entities in the Order package.

The request can be tracked and turned into a requirement which can be used to create a Work Effort (a task, for instance; see the Work Effort section below) that will satisfy the requirement and fulfil the request. Once a requirement is made, a quote can be produced which, if accepted by the customer, can be used to create an order. Once an order is fulfilled an invoice can be created from the order. Invoices are part of the Accounting entity package described below.

Orders consist of an Order Header and any number of Order Line Items and Adjustments that describe the detail of the Order. There are various pieces of information related to an Order that can be associated with either the header or an individual or multiple line items of an order. Examples include the shipping destination and shipping preferences of an order which may be the same for all line items, or may be different for each one.

Adjustments are used to contain information about things that change the price of an order that are not actual goods or services sold or purchased. Examples include taxes, shipping, discounts, surcharges, and so

forth. An adjustment can be either a flat amount, a flat amount per quantity, or a percentage of the subtotal of the entire order or the line item it is associated with. Tax and shipping are handled as special cases and each adjustment can specify whether or not it should be included in the sub-total for tax and/or the sub-total for shipping.

Payment preferences can be tracked as an order is created to automate payment once an invoice is created. This is especially useful for payment by credit card or other electronic means. If no payment preferences are specified then a standard invoicing and billing process can be used.

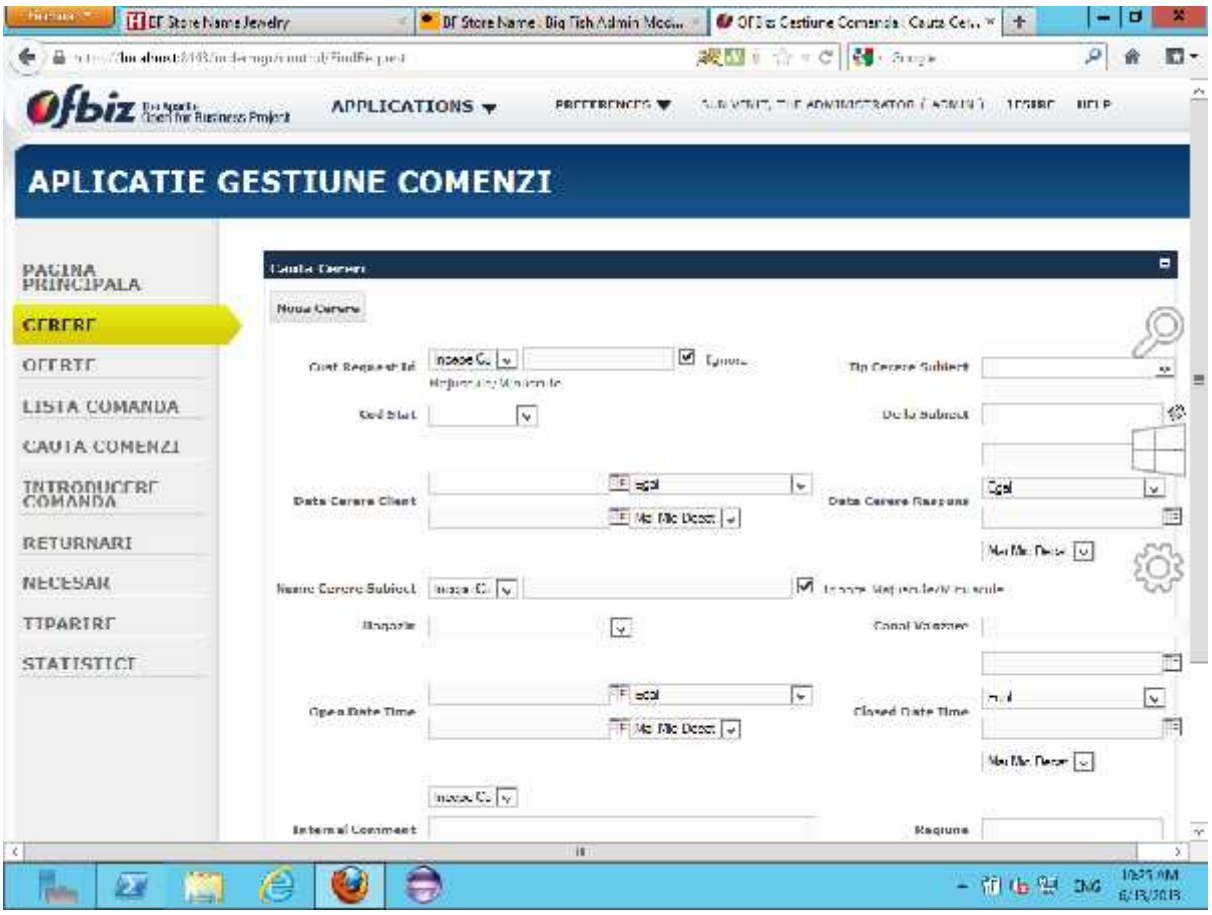

Figure 5 – Ofbiz Order

Facility

A Facility is a building or other physical location. Examples include warehouses, stores, office buildings, individual rooms in a larger facility, loading docks, et cetera. Generally a facility will have Contact Mechanisms associated with it such as a postal address or a phone number.

Facilities can be grouped into Facility Groups which in turn can be contained in other Facility Groups. Examples of groups include store chains, districts, regions, and special groups used for marketing or the pricing of products.

Inventory Items can be associated with a facility and even a specific location within a facility. Facility Locations can be tracked and managed independently of the Inventory Items they contain for easy management of warehouse spaces and easy location of specific Inventory Items.

Parties can be associated with Facilities to represent where a person works, which organization controls or operates the facility, who manages the facility, et cetera.

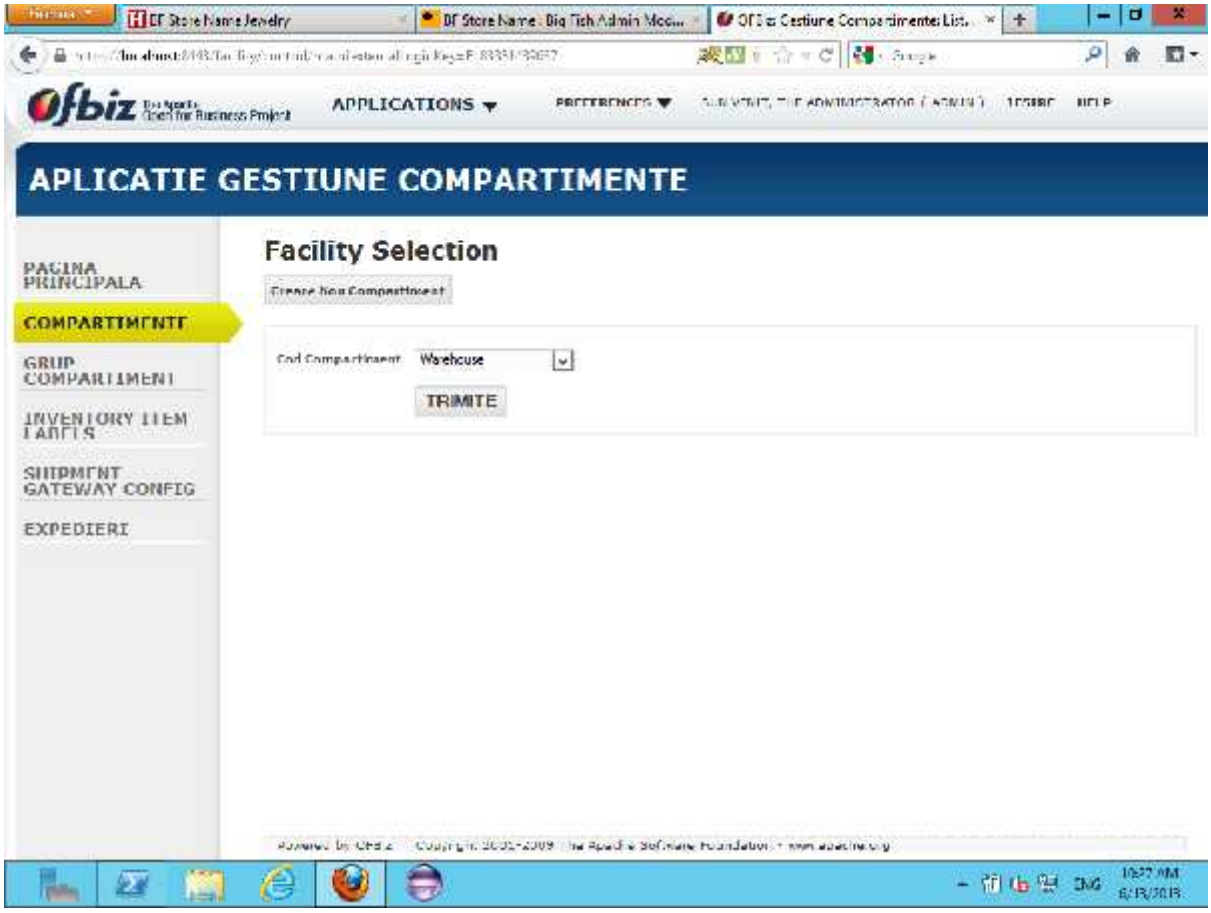

Figure 6 - Ofbiz Facility

Shipment

The Shipment entities are used to track incoming and outgoing shipments and to issue items from inventory or receive items into inventory.

A Shipment consists of multiple Shipment Items each of which, like an Order Item, represents a certain quantity of a specific Product. When Shipments are received they can be reconciled with a purchase order and that information is tracked in the Shipment Receipt entity.

When an Inventory Item is issued for an outgoing shipment it is associated with a Pick List that can be prepared from multiple Shipments to more efficiently route the picking path in the warehouse or other facility.

Shipment Packages can be created which represent a single box or shipping unit. A single box can contain multiple Shipment Items, even items from different Shipments assuming that they are going to the same destination.

The Shipment Route entities are used to split up a Shipment's journey into multiple route segments. One route segment for a Shipment could be a commercial carrier while another could be a private carrier or a company owned truck.

Accounting

The Accounting entities are organized according to old age and generally accepted principles such as double-entry accounting, a General Ledger with hierarchical accounts, journals and posting of transactions and corresponding entries. The structure is primarily based on the OMG GL standard and the work that was done on an AR/AP extension of the OMG GL standard. This correlates well with other standards such as ebXML and OAGIS.

The Accounting entities are structured such that accounts for multiple organizations can be managed. The multiple organizations could be multiple companies, or departments or other organizations within a company. Each Organization can have various GL Accounts associated with it so that it can operate with its own subset of the Master Chart of Accounts. Each Organization can also have its own set of Journals for flexibility, even though the use of Journals should be as minimal as possible in

favour of allowing the system to automatically create and post transactions based on business events triggered by standard procedures and documents such as purchase and sales orders, invoices, inventory transfers, payments, receipts, and so forth.

There are also entities in place for budgeting and the reconciliation of budgets against actual GL Account balances for a specific fiscal period. There are also entities used to track custom Fiscal Periods and other entities to keep summary results for accounts in specific periods.

Entities to track Fixed Assets are also part of the Accounting entity package. This includes depreciation information in addition to maintenance, scheduling of Fixed Assets (along with the Work Effort entities), and so forth.

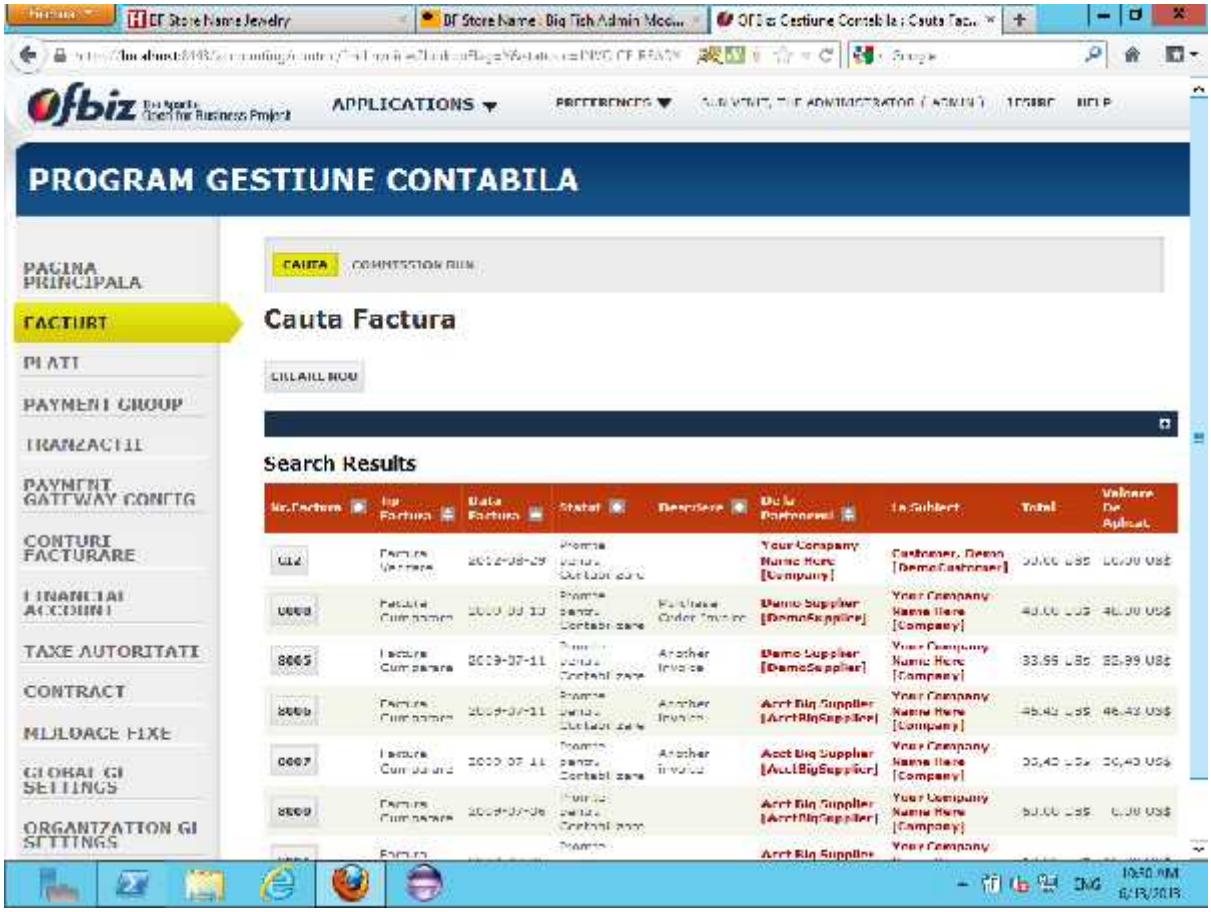

Figure 7 - Ofbiz Accounting

Marketing

The Marketing entities are used to track information about Marketing Campaigns and related information such as Contact Lists (mailing, email, or calling lists) and Tracking Codes. Tracking Codes are primarily used in automated systems to keep track of where a customer came from and can be used for commission purposes in addition to analysing the effectiveness of specific Marketing Campaigns including advertisements, partnerships or affiliations, and so forth.

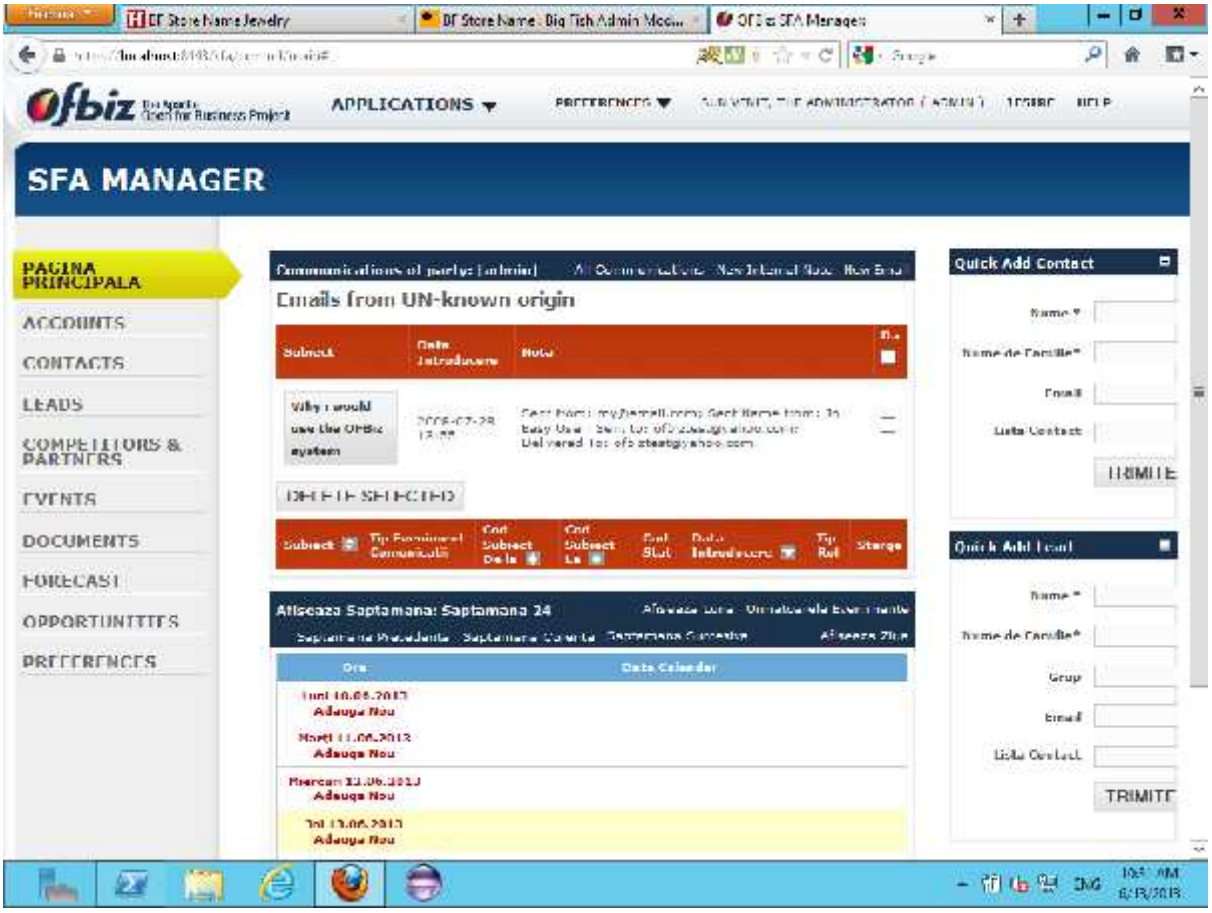

Figure 8 - Ofbiz Marketing

Work Effort

A Work Effort can be one of many things including a task, project, project phase, to-do item, calendar item, or even a Workflow Activity.

Notice: OFBiz uses an Event Driven Architecture (EDA) and ECAs (SECA, EECA, MECA) are used in OFBiz to drive the Workflow. ECA is the acronym of Event Condition Action. SECAs are for Services (triggered on services conditions), EECA are for Entity (triggered on entities conditions), MECAs are for Mail.

There are also entities in the Work Effort entity package for keeping track of timesheets and the pay rate of specific Parties performing Work Efforts. In addition various other resources can be tracked such as Fixed Assets, Facilities, et cetera.

When Work Efforts are used for manufacturing or other modification of Products the Products and Inventory Items consumed and produced by the specific Work Effort can be tracked.

Human Resources

The Human Resources entities are used to keep track of positions, responsibilities, skills, employment, termination, benefits, training, pay grades and payroll preferences, performance reviews, resumes and applications, and other Human Resources related information.

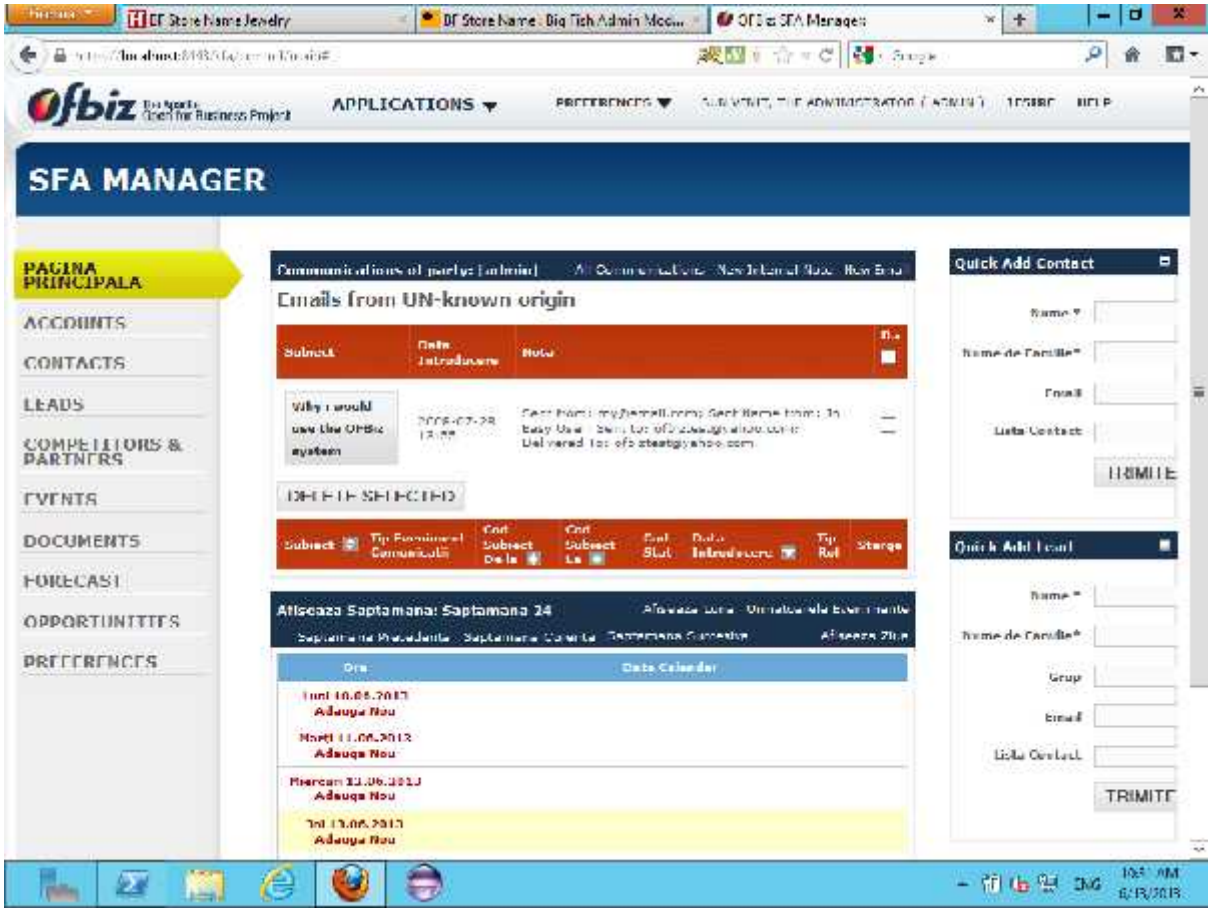

Figure 9 - Ofbiz Human Resources

Architecture and System Organization

Architecture is really just a fancy word for the organization and composition of application components. There are many different "tools" available as part of Java, J2EE and the OFBiz Core Framework that can be used together in order to efficiently and effectively organize data and business logic, to provide interfaces to other systems, and to create user interfaces for humans to interact with the system.

Entities and Services

The most basic components in OFBiz are Entities and Services. An Entity is a relational data construct that contains any number of Fields and can be related to other entities. Basic entities correspond to actual database structures. There is also a type of entity called a "view-entity" that can be used to create a virtual entity from other entities to combine sets of fields by joining other entities together. These constructs can be used to summarize and group data in general and prepare it for viewing or use in a program.

In usual architectures the data or persistence layer alone consists of hundreds of thousands or millions of lines of code that must be maintained as the system is developed and customized. With the Entity Engine this is all distilled into only thousands of lines of XML data definitions that drive an easy to dynamic API. Even less-experienced programmers can become productive with this tool in a few days without learning SQL.

There is the "Web Services" buzz that has spread to every corner of the software industry. The OFBiz system not only uses the service pattern to communicate with other systems, it also uses the service pattern inside the system to provide a clean and easy to use facility for creating and running business logic components.

A Service is a simple process that performs a specific operation. A service definition is used to define the input and output parameters that the service consumes and produces. Data passed to the service can be automatically validated before the actual logic is called using this definition. After a service is run the results can be validated in the same way.

Other services can be run automatically at different points of the running of a service by

using Event-Condition-Action (ECA) rules to denote what other services should be called and under what circumstances. This allows logic to be extended without modifying the original logic and allows the system to be organized cleanly such that each service performs one simple, specific task.

Services can be implemented in a number of different ways to make it easier for engineers to match the tools available to the task at hand. It also makes it easy to keep track of the logic components in the system which may exists in hundreds of different files and even on different computers used inside the company or computers of a partnering company.

Web Framework and XML Mini-Languages

There are many useful tools in the OFBiz Core Framework that address points of difficulty in web, client-server and peer-to-peer based enterprise applications. A tool exists for simplifying the use of flat data files making it easier to integrate with legacy systems. Various tools exist for organizing and structuring web-based content and applications with a flexible separation of logic and presentation. These tools along with the standard J2EE and other Java tools make it easy for the system to communicate with human users and other systems.

To make it easier to use these Core Framework tools another component called an XML Mini-Language has been created that allows an engineer or other user to create an XML file with simple, well-defined instructions that the system can understand and execute. Expressing logic in a Mini-Lang "simple-method" often requires only one third as much code as an equivalent Java method and is much easier to read and modify for semi-technical users. Processing form input, composing services into larger services, and interacting with data in the database are common things that can be done easily with this tool.

CONCLUSION ON APACHE OFBIZ SOLUTION

The Open for Business Project is a collaborative effort involving a large group of users and developers and is moderated by a small central team. The applications and framework components are being used in a wide variety of businesses and applications and are customized heavily in many circumstances to meet the needs of the organizations using the software.

This is only possible and can only operate efficiently and effectively as an open source project. This is in sharp contrast to the style of most commercial vendors that restrict many possible uses of and extensions to their software through design limitations or licensing restrictions in order to extract a greater profit from the use of the software. Much of the early commercial software was much more flexible and open and less restrictive than many modern packages. As that trend continues more developers and users of software start to consider a more trusting alternative.

While those who contribute to the project, including the moderators, choose not to force others who use it to pay them, they can benefit in many other ways. Often the user will find that the system is so close to what they need that a small amount of effort can change the system to be perfect for them. If that effort is contributed back to the community others in a similar situation can help improve and maintain it. The contributor can freely upgrade his own system with other improvements in the main project without having to worry about changes that might conflict with his addition.

In general OFBiz is meant to provide so much value that the user will be willing and happy to contribute something back and keep the effort going. This again is in sharp contrast to the commercial approach of forcing users to pay for every right of use or modification related to the software. The open source approach enables information sharing and that information can be leveraged by all in the community to improve the use of the software for their own needs.

So, for the moderators of the project: what do they get for all of the time spent designing frameworks and applications, writing code, and answering questions? It goes back to the basic law of the harvest: you reap what you sow. The more value something provides, the more it is worth.

References

[1] Wikipedia: http://en.wikipedia.org/wiki/List_of_ERP_software_packages
[2] Apache Ofbiz Documentation: http://ofbiz.apache.org/documentation.html
[3] All the print screens that are presented in this article are from my version of Ofbiz installed on my development server.