

# Accounting for Developers

Jeannie Dobney & Anne Ristau

## Introduction

The technical team often hears functional people using a lot of accounting jargon. This presentation will provide background and context to help understand that 'bean-counter' jargon.

Covering the same topics as an "Accounting for non-Accountants" course, this presentation will also relate those topics back to e-Business suite development tasks and provide a simplified introduction to the key accounting functionality of key Financial Applications.

## Approach

The Approach we take in this paper is to take 3 passes through the same material:

- In the first pass we will describe our example of a business and its transactions: Joe's Bakery
- In the second pass we will explain how Joe uses the Oracle eBusiness Suite to store those business transactions
- And finally we will explain the accounting entries generated by each transaction.

We hope this approach makes the material easily understood even if you are starting from no understanding of accounting at all. Note that the accounting transactions are only indicative – they are used to illustrate accounting concepts. Readers who are interested in definitive accounting should refer to the Useful References Appendix at the end of the paper.

## Introducing Joe's Bakery

Joe has decided he wants to work for himself and so he is starting a new business: a bakery.

He has found a nice location on a busy street and he is sure he will be able to sell lots of bread from there. He has taken his business plan to the bank and the manager is willing to lend him the money to cover the bond for the lease on the shop.

Joe's Aunt Mabel died recently and left him some money – as it turns out it is just enough money to buy bread making machinery.

So Joe orders the bread making machinery and the flour and other things he needs, including a sign for the front window and some display cabinets.

Joe opened last Monday and sure enough trade was brisk from the start.

Every afternoon he has been taking cash to the bank and has even been able to consider hiring his cousin Fred to help at the counter.

He has been delighted to find that every day he has sold more than the day before – and so he has to order more flour and some bigger bowls already. After lunch each day he now checks to see if he has enough flour for the next day - and to order more if it's required! Fortunately the day's takings have always given him the cash required for the next delivery of flour – because the mill only does rush deliveries C.O.D. (cash on delivery).

Joe's friends are asking how much profit he's making – so he has decided that he needs the Oracle eBusiness Suite to store and analyze his accounting information – and to allow him to hire and pay his cousin Fred when that happens.

Joe has 2 friends who've just started a consulting company the *Ristau Dobney Huge Fees Alliance (RDHFA)*, and in exchange for some of the great pizza slices Joe makes, they've mapped his business requirements to the Oracle software to help Joe understand how Oracle can support his business. The next section describes what they recommended.

## How Joe will use the Oracle eBusiness Suite Software

RDHFA recommend Joe implement (set-up) the following Applications for Joe:

The **General Ledger** is the central repository of accounting information, it holds both summary data (account balances) and detail level (transaction) data.

The key to setting up the General Ledger is the creation of a Chart of Accounts. To do this the Accounting Flexfield is used.

There are 2 types of Flexfields: Key (KFF) and Descriptive (DFF):

*Descriptive Flexfields* are columns appended to standard tables that allow storage of user defined information. The available columns are called AttributeN (where N is a number) and in the Applications the presence of these columns is represented by a single character field labelled with square brackets i.e. [ ].

*Key Flexfields* store fundamental Application data in multi-segment fields which are configured during the implementation. The data domain for each KFF segment is defined by a value set. These values are then combined and each combination is given a key, which other Applications reference as a foreign key.

The design of the *chart of accounts* is a major focus of any implementation:

- how many segments are required, e.g. company, cost centre and natural account
- what sort of data should be stored in each
- which values are required in each segment.

The design of the chart is so critical because it determines what sort of reporting is possible – and the major driver of most implementations is reporting in some form i.e. producing reports to comply with tax and corporate regulations or for management control purposes.

A typical account code combination may look like this: 01-110-5231.

This value might represent the following:

01 is the balancing segment value – at this level an organisation produces a balance sheet (which is the ultimate financial report). It might represent a company or perhaps an agency fund in the public sector.

110 is a cost centre, it might be the purchasing department or the sales department. At this level the focus is likely to be management reporting e.g. a manager will be responsible for monitoring how much is spent each month for this specific business area.

5231 is a natural or true account. The organisation requires one value for each type of account value they need to track. So for example this might be Stationery expense. Having a separate account for stationery means that the organisation can track actual expenditure against their forecast budget for this item.

RDHFA recommend Joe set up his Accounting Flexfield as follows:

	Company Segment	Cost Centre Segment	Account Segment
Defined	A	100	5555
Flex	B	200	6666
Values:	C	300	7777

The first segment has a qualifier which tells the system that it is the *balancing* segment. This means that we will produce a balance sheet (explained later) at this level. For now Joe only needs one value ('HQ') there but this segment is mandatory – and if Joe decides to have more branches later, he can add a value here for each store.

The second segment holds the Cost Centre (again mandatory) which will allow organizations to separate transactions into areas like sales, research & development, training and corporate overheads. In Joe's case the values recommended are:

- 000 – No Cost Centre (for expenditure to be excluded from the following)
- 100 – Bread and Savory Bakery Goods
- 200 - Cake and Sweet Bakery Goods
- 300 – Ready to Go Bakery Goods (like the excellent pizza and ready to eat sandwiches)

The final segment is the so called natural account and the values in it will be explained in more detail later. However RDHFA recommend the following structure:

Account structure:

Assets	1000
Liabilities	2000
Equity	3000
Revenue	4000
Expenses	5000

Within each section there will be a hierarchy of values, so for example within the Expenses, RDHFA recommend something like the following sample values:

General Expenses	5100
Bakery Consumables	5110
White Flour	5111
Whole meal Flour	5112
Office Consumables	5120
Stationery	5121
Labor Costs	5200
Salaries	5210
Salaries paid	5211

The structure shown above has both:

- Parent values (end in a zero) and allow summarization in reports.
- Posting values (end in non-zero) and used for assigning actual transactions.

The account numbering convention normally helps accountants group these values into hierarchies for convenient reporting. In the example given above all expenses might start with 5nnn and Stationery expense (5231) might 'roll-up' to a parent account like 5200 General Expenses, which in turn might roll up to 5000 All Expenses.

Later we'll see how these values are combined to help Joe analyze his financial information and make business decisions.

RDHFA have also recommended that Joe implement the following Procurement Applications:

Accounts Payable – This is the Application used to record supplier invoices and issue checks or EFT payments. Joe will need both of these functions to process the bills he receives from his suppliers.

iExpenses – Although Payables also supports the entry of employee expense reports, this type of functionality is more typically provided by the web-based iExpenses Application. Joe (and Fred after he joins) will have to be reimbursed for expenses they incur on behalf of the organization, so they will also need this Application.

Purchasing – (sometimes also called core Purchasing to distinguish it from iProcurement); this is the Application used to create Purchase Orders and it owns the Supplier (or Vendor) tables. It can also create Requisitions and Receiving transactions, however these days most people use the web-based iProcurement tool for both these functions.

Purchase Orders are important procurement documents because they represent a legal commitment to pay for the requested goods and they normally bear the organization's logo and possibly some legalese about Terms and Conditions.

Receiving Transactions support 2 purposes:

Firstly to record that goods or services have been received, because most organizations do not want to pay until that has occurred.

Secondly for procurement into Oracle Inventory (to enter the correct warehouse location etc for the goods). Joe does have an 'inventory' of flour and other consumables but he can keep an eye on these visually at this stage – he doesn't need Oracle's sophisticated supply chain management yet – so he will only use Oracle Purchasing to create Purchase Orders and to maintain records about the flour mills and his other suppliers.

iProcurement – supports web-based Requisitions and Receiving transactions  
Requisitions are employee requests for goods or services. IProcurement uses Workflow to route them to supervisors before they are converted into Purchase Orders. Obviously Joe's business will run more informally so this tool will not be implemented for Joe.

So that takes care of the money going out of Joe's bakery, now for RDHFA's recommendations for how Oracle can support the money coming in to Joe's bakery.

Accounts Receivable – is the Application used to maintain customer records and to create invoices to customers (like Joe's weekly bill to the corporation that has a standing order for muffins). It is also used to receipt incoming money – but normally not for retail situations, so he will probably need a Point of Sale (POS) system which he'll connect to Receivables via the iPayment Application (or the new Payment engine in Release 12).

Order Management – this Application can be used to process incoming order and to interface those orders to Receivables where they become Invoice Transactions; but this functionality is more suited to businesses that have call centers processing and shipping orders, so Joe will not need this Application

Cash Management – this application has 2 functions which Joe needs.

Firstly, Joe needs to reconcile his bank account to the payments he made from Payables and those that he recorded as coming in via Receivables.

Secondly Joe needs to forecast his cash flow i.e. to make sure that the money coming in is sufficient to cover all his outgoing payments.

There are a couple of other Applications RDHFA's also recommend Joe consider to support his bakery:

Fixed Assets – this Application stores and supports the accounting treatment required for Joe's assets. The accounting definition of an asset will be provided later in this paper, however we will see that Joe's bakery machinery is considered an asset.

Projects – the Projects suite of Applications are normally used to support businesses like consulting companies that have lots of 'projects' i.e. activities with defined time boundaries and deliverables (specific outcomes). Joe doesn't need this type of functionality so these Applications.

Human Resources and Payroll – Joe will be able to record details about his employees, their skills and entitlements in the Oracle Human Resources Application – and he can generate salary payments from the Payroll Application.

In the next section we'll take a more detailed look at how transactions flow through the Oracle eBusiness Suite from a more general perspective.

## **eBusiness Suite Process Flows**

So now that we know which Applications Joe will use, let's review how transactions flow through them.

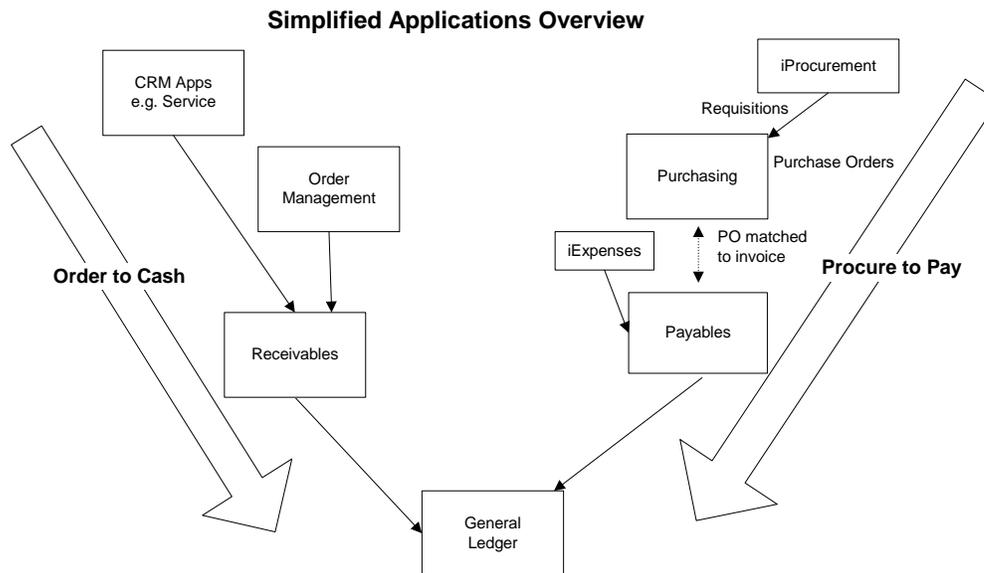
The eBusiness Suite is a set of individual products (Applications), these Applications may be thought of as both a collection of system functionality (required for a subset of functionality) and / or an accounting sub-ledger i.e. a specialised accounting function.

Transactions flow through the Applications as described below.

The heart of the eBS Financials suite is the *General Ledger*, 2 main "transaction arms" flow into the General Ledger:

- The *Purchase to Pay* process flow leads from requisitions goods or services, through payment to suppliers for those goods or services to updating the relevant GL account balance. In simplified terms this represents money flowing *out* of the organisation.

- The *Order to Cash* process flow originates in customers requesting goods or services, being invoiced and paying for those goods or services, to updating the relevant GL account balance. In simplified terms this represents money flowing *into* the organisation.



The heart of the manufacturing and distribution systems is the *Inventory* application, where stock items are defined, and stock balances are updated as new goods are requisitioned, sold or manufactured.

The *Human Resources* suite of products consists of 2 core Applications: Human Resources and Payroll plus a set of web-based Applications including things like iRecruitment.

An Appendix at the end of this paper describes the key functionality of each Application and related business and system processes in more detail. The intended audience is technical staff and these descriptions are intended as an introduction to business processes, rather than comprehensive descriptions.

## Joe's Accounting Transactions

Having seen how business transactions are handled by the Oracle eBusiness Suite, the next things to consider is the accounting transactions generated by those transactions.

We noted above that the General Ledger stores 2 kinds of data:  
 transaction data i.e. *journals* and  
 cumulative account *balances*.

An account balance is stored for each account code combination (in the chart of accounts), and for each period.

*Transactions* use a convention called double entry book-keeping, which means that each transaction has at least 2 impacts. Those impacts must *balance* (or cancel one another out).

Consider the example of purchasing raw materials like flour for Joe's bakery:

\$1,000 of raw materials is represented as follows:

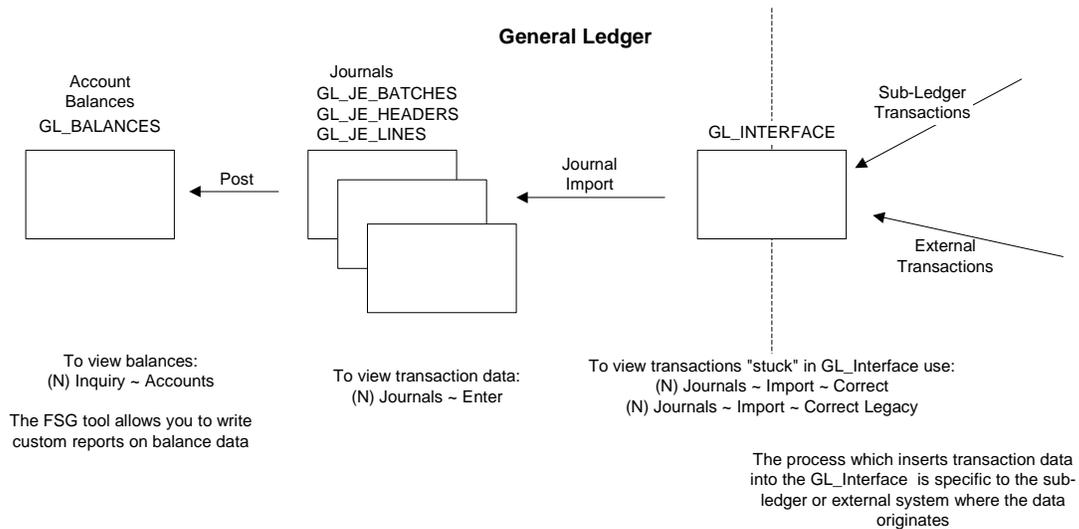
Debit \$1,000 Raw materials expense  
 Credit \$1,000 Cash-at-bank

In this example the Debit increases the Expense balance and the Credit reduces the Cash balance. (Whether a debit increases or decreases a balance depends on the type of account).

In some cases both sides of a transaction may be explicit, for example in manual adjustments in GL where a cost is being moved from one account to another. In some cases, however, it might seem like the transaction has only one side, for example, when you look at an invoice in Payables; this is because the other side of the transaction is always recorded in the same account: a control account typically used for reconciliation.

*Balances* represent the cumulative result of transactions. So for example assume that in addition to the transaction shown above, the organisation also paid wages of \$5,000 in a specific period. If the organisation previously had a balance for Cash-at-Bank of \$10,000, it will have a new balance of \$4,000 after the new transactions are "posted" (i.e. \$10,000 – [1,000 + 5,000]).

In the Oracle eBusiness system, most transactions are initially recorded in sub-ledgers (i.e. specialist Applications catering to specific types of activities like procurement) and flow from there into the General Ledger, as shown below:



We also noted above that an account balance is stored for each account code combination (in the chart of accounts), and for each period. However, before we continue, it is necessary to understand a bit more about that third segment, the *account* segment.

Natural accounts must be "qualified" as 1 of 5 types:

*Revenue* – i.e. fees and payments made by customers to the organisation

*Expenses* – i.e. payments made by the organisation to suppliers

(Profit is Revenue minus Expenses)

*Assets* – i.e. organisational expenditure for which accounting regulations deem that the purchased item delivers value over a period of time (e.g. years). Such expenditure must be "expensed" over the life of the asset (this is called Depreciation).

*Liabilities* – i.e. debts owed by the organisation but not immediately due for payment. Examples include bank loans, due for payment several years in the future or salary related commitments like long service bonuses being accumulated for employees.

*Equity* – i.e. the net value of the organisation represented (and equivalent to Assets minus Liabilities).

Having understood all this, let's look at Joe's first week of business (described in one of the first sections of this paper) let's look at the accounting transactions which would have been created.

Joe opened a bank account and put the money from Aunty Mabel's estate into it as his 'seed capital'.

Because that is Joe's money, it is recorded as a Director's loan using a liability account:

Dr Cash at Bank \$50,000  
Cr Director's loan \$50,000

Joe's bank manager loaned him money to cover the bond for the lease on the shop.

This loan is recorded as a liability and initially increases his bank balance:

Dr Cash at Bank \$20,000  
Cr Bank Loan \$20,000

Then the lease bond is paid by bank cheque, reducing the bank balance again:

Dr Lease Security \$20,000  
Cr Cash at Bank \$20,000

Probably Joe will record these transactions as manual journals via GL, because they occur 'outside' of Oracle.

Then Joe purchases his bakery machinery.

The company which sells him the machinery, won't accept his order without a Purchase Order on his letterhead, so Joe creates a Purchase Order and sends it to the store.

This transaction in Oracle creates no accounting entry, unless you are using Public Sector Applications in which case it creates an encumbrance transaction.

Alternatively if Joe was using Oracle Projects, he would see a 'commitment' in Projects for the amount of the machinery.

The machinery arrives and so does the supplier's invoice. When Joe enters this transaction, he will see his first accounting entry generated in Payables and interfaced to GL:

Dr Plant and Equipment \$25,000  
Cr AP Control \$ 25,000

Notice that the debit is to an *asset* account because the bakery machinery cannot be expensed in the current period. This invoice line will be interfaced to Oracle Assets and there becomes an asset.

When the check is generated from Payables to pay for the machinery, the following transaction is created by Payables and interfaced to GL:

Dr AP Control \$ 25,000  
Cr Bank \$ 25,000 (or a Cash Clearing Account if using Cash Management)

When Joe purchases flour and other consumables, they will be 'expensed' (i.e. debit an Expense Account rather than an Asset account).

Dr Plain White Flour \$2,000  
Cr AP Control \$ 2,000

Payment transactions are identical i.e. for asset and expensed procurement.

Accounts Payable interfaces all these transactions to General Ledger using a concurrent process called the *Payables Transfer to General Ledger*. GL pulls them into its transaction / journal tables using the *Journal Import* process and updates the balance data using the *Posting* process.

Joe will use Oracle Assets to track his assets. He now has the following assets on his asset register (total cost \$35,000):

Bread making machinery  
Front window signage  
Display cabinets.

Each month Depreciation Expense will be generated, this gradually expenses the cost of the asset over the life of the asset.

On the revenue side, we know that since Joe opened trade has been brisk. Each day he gets entries like the following from his Point of Sale system:

Dr Cash at Bank \$5,000  
Cr Revenue \$ 5,000

Each entry includes a cost centre value which will help Joe track and analyze which of his products is most profitable. So the sale of a cream bun actually goes to the following account code:

HQ – 200 – 4111  
that means:  
Branch = HeadQuarters  
Cost Centre = Sweet Bakery  
Account = Revenue (General)

So at the end of the week, Joe can now run his Income Statement to answer his friends questions about how profitable he has been:

*Joe's Bakery Income and Expenditure (HQ)*

*Revenue*

Revenue (General)	\$25,000
Revenue (Orders)	<u>\$ 5,000</u>
Total Revenue	\$30,000

*Expenses*

Plain Flour	\$ 6,000
Wholemeal Flour	\$ 2,000
Other Ingredients	\$ 4,000
Rent	\$ 8,000
Wages (accrued)	<u>\$ 5,000</u> (*)
Total Expenses	\$25,000
Profit / Loss	\$5,000 (16%)

(\*) An accrual is an entry which is made to represent a cost you know you had during a period but for which there is not yet an actual transaction. In this case that means Joe has worked in the bakery and earned wages but the wages check hasn't been written yet.

Joe is reasonably pleased with this result from his first week, but he wonders if he really has enough money in the bank to actually pay himself – so he looks at his Balance Sheet (another FSG which RDHFA have set up for him).

*Joe's Bakery Balance Sheet (HQ)*

*Assets*

Plant & Equipment	\$35,000
Cash at Bank	\$25,000
Lease Security	<u>\$20,000</u>
Total Assets	\$80,000

*Liabilities*

Bank Loan	\$20,000
Director's Loan	\$50,000
Salary Accruals	<u>\$ 5,000</u>
Total Liabilities	\$75,000

*Equity*

Current Profit / Loss	<u>\$ 5,000</u>
Total Equity	\$ 5,000 (Note the equation: Assets + Liabilities = Equity)

Joe sees that he does have enough money in the bank to pay himself and for next week's flour deliveries, although he decides to wait another week before he hires his cousin Fred.

Joe has now seen how those 2 FSG reports provided the information he needed to make his business decisions. So Joe now decides to run the *Profit and Loss by Cost Centre Analysis* (another FSG which RDHFA have set up for him). This is what it shows him (factored by thousands)

<i>Cost Centre</i>	<i>Revenue</i>	<i>Expenditure Ingredients</i>	<i>Expenditure Labor, Rent etc</i>	<i>Profit / Loss</i>
100 – Savory	17	7	7	3
200 – Sweet	10	2	4	4
300 – Ready to eat	3	3	2	(2)
<b>Total</b>	<b>30</b>	<b>12</b>	<b>13</b>	<b>5</b>

Joe sees that

his savory goods have a high turn-over but a relatively low profit margin,  
his sweet goods have a low turn-over but a high profit margin  
but that his ready to eat items are running at a loss.

(probably because RDHFA are allowed free pizza during their configuration...)

## **Conclusion**

This paper has traced a simple business scenario from the everyday activities of Joe's Bakery through its accounting transactions to the balance sheet. The objective has been simply to provide a conceptual introduction to help technical people understand what accountants do. Like all high level introductions it includes lots of assumptions and generalizations but the authors hope it helps provide technical people an introduction to accounting.

## **About the Authors**

Anne Ristau developed the idea of Joe's bakery when she developed and taught the Technical Foundation course for Oracle University. Anne has regularly presented papers for Oracle users since the first OAUG conferences and is currently a director of the OAUG. Anne can be contacted by email at [RistauA@crlcorp.com](mailto:RistauA@crlcorp.com).

Jeannie Dobney has also been a supporter of the OAUG for over a decade and has regularly presented papers at conferences over that period. Jeannie is solely responsible for any accounting errors you may find in this paper and if you wish to point them out, you can contact Jeannie by email at [jdobney@bigpond.com](mailto:jdobney@bigpond.com).

## **Appendix: Useful References:**

There are many useful references on the Internet, for example:

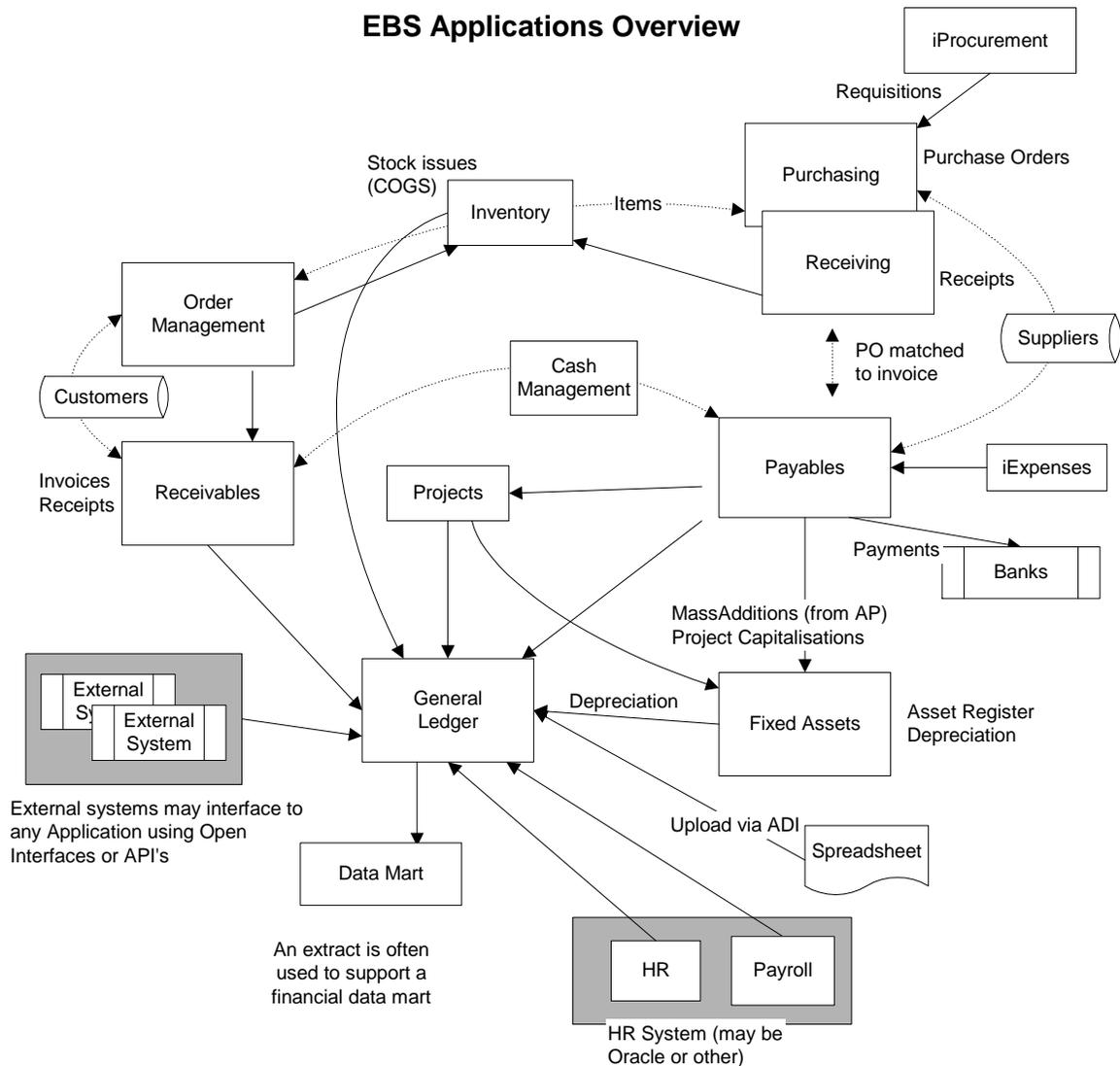
<http://www.accountingcoach.com>

<http://www.dwmbeancounter.com/tutorial/Tutorial.html>

<http://www.simplestudies.com/accounting/lec/accounting0101.htm>

## Appendix: Application Detail

The following diagram reflects a more sophisticated view of a typical Applications architecture.



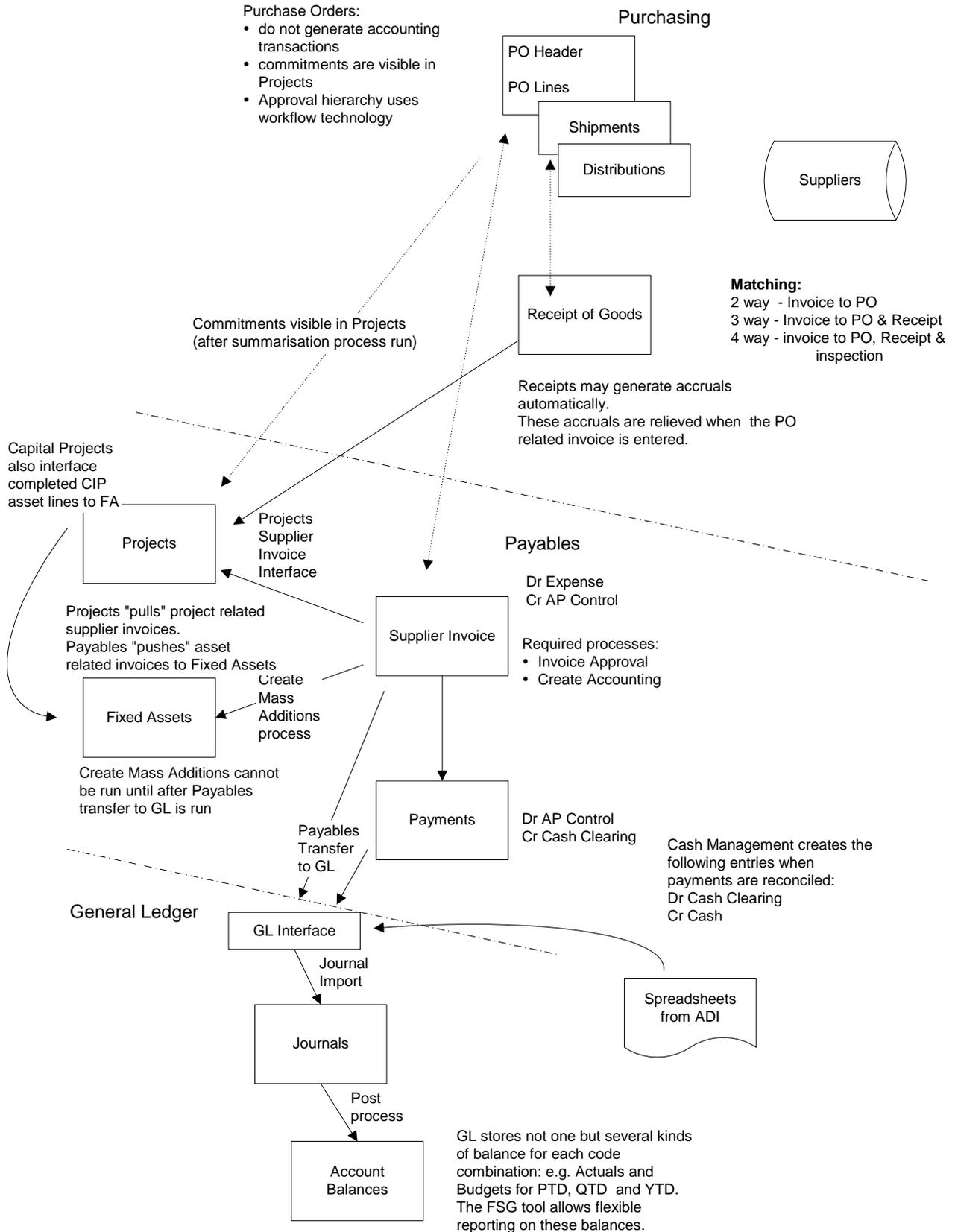
The exact process flows vary for each site depending on how the Applications have been implemented. Factors which determine different configurations include the following:

- The implementing organisations specific needs. For example a company with no warehouse or distribution needs will not choose to implement Oracle Inventory (except as required to support Purchasing activity).
- Pre-existing systems may need to be integrated with the Oracle eBS system. This may be done via an open interface or API scheduled to run in the background cross updating both systems as required.
- Specific business processes for example a public sector organisation may require 'encumbrance' (commitment) accounting at the time a purchase order is created whereas a private organisation may not require actual accounting entries until a supplier invoice is entered.

The next diagram is a more detailed view of a subset of these tables showing the accounting entries and processes used by this sample implementation. (Remember that the exact details will be implementation specific).

# Oracle Applications Purchase to Pay with Projects

- Purchase Orders:
- do not generate accounting transactions
  - commitments are visible in Projects
  - Approval hierarchy uses workflow technology



## General Ledger

The heart of the Financial system is the General Ledger (GL) and that the sub-ledgers feed data into it.

There are 2 key data stores in GL:  
transaction data i.e. *journals* and  
cumulative account *balances*.

An account balance is stored for each account code combination (in the chart of accounts), and for each period etc.

The design of the *chart of accounts* is a major focus of any implementation:  
how many segments are required, e.g. company, cost centre and natural account  
what sort of data should be stored in each  
which values are required in each segment.

The design of the chart is so critical because it determines what sort of reporting is possible – and the major driver of most implementations is reporting in some form i.e. producing reports to comply with tax and corporate regulations or for management control purposes.

A typical account code combination may look like this: 01-110-5231.

This value might represent the following:

01 is the balancing segment value – at this level an organisation produces a balance sheet (which is the ultimate financial report). It might represent a company or perhaps an agency fund in the public sector.

110 is a cost centre, it might be the purchasing department or the sales department. At this level the focus is likely to be management reporting e.g. a manager will be responsible for monitoring how much is spent each month for this specific business area.

5231 is a natural or true account. The organisation requires one value for each type of account value they need to track. So for example this might be Stationery expense. Having a separate account for stationery means that the organisation can track actual expenditure against their forecast budget for this item.

Natural accounts must be “qualified” as 1 of 5 types:

*Revenue* – i.e. fees and payments made by customers to the organisation

*Expenses* – i.e. payments made by the organisation to suppliers

(Profit is Revenue minus Expenses)

*Assets* – i.e. organisational expenditure for which accounting regulations deem that the purchased item delivers value over a period of time (e.g. years). Such expenditure must be “expensed” over the life of the asset (this is called Depreciation).

*Liabilities* – i.e. debts owed by the organisation but not immediately due for payment. Examples include bank loans, due for payment several years in the future or salary related commitments like long service bonuses being accumulated for employees.

*Equity* – i.e. the net value of the organisation represented (and equivalent to Assets minus Liabilities).

A numbering convention normally helps accountants group these values into hierarchies for convenient reporting. In the example given above all expenses might start with 5nnn and Stationery expense (5231) might ‘roll-up’ to a parent account like 5200 General Expenses, which in turn might roll up to 5000 All Expenses.

The General Ledger stores a *balance* for each account code combination (e.g. 01-110-5231) in the GL\_Balances table and Oracle provides a report writing tool specifically for reporting from this table (i.e. the Financial Statement Generator or FSG tool).

Account balances are updated from each day’s transactions and represent the cumulative result of transactions. The update process is called Posting.

Most transactions are initially recorded in sub-ledgers (i.e. the specialist Applications catering to specific types of activities like procurement or inventory management) and flow from there into the General Ledger, through the GL Interface.

All transactions regardless of their origin may be viewed as Journals in the General Ledger once they have been imported successfully (by using the concurrent process Journal Import).

## Purchasing

Requests for new goods or services begin as *Requisitions*. These may be paper based or entered directly into a system (like Oracle). The key from management's perspective is to be able to authorise spending before it occurs. iProcurement and Purchasing provide workflow processes to forward requisitions to an approval hierarchy reflecting the organisations delegations (i.e. which managers are authorised to approve which specific levels of spend).

Requisitions may also be *internal* (rather than purchase requisitions) i.e. if the organisation maintains a warehouse (represented in Oracle Inventory), an internal requisition is a request to release goods from the warehouse. Oracle provides concurrent processes and workflows which enable such requisitions to be forwarded to Order Management for fulfilment.

Oracle *iProcurement* provides an HTML based interface which can be widely deployed within organisations (i.e. beyond Purchasing specialists) to submit requisitions and record the receipt of goods and services. Recently some organisations use iProcurement to access (punch-out to) external catalogues of goods.

Approved requisitions are placed by the system in a requisition pool and from there buyers (purchasing specialists) collate them onto purchase orders; this process is called *AutoCreate* in Oracle Purchasing. In organisations which do not have a centralised purchasing function, the AutoCreate process may be automated using a modified workflow.

*Purchase Orders* (PO's) are significant documents because they legally commit an organisation to the procurement specified on them. They may be emailed, faxed or printed from Oracle Purchasing for external suppliers. In addition to standard PO's, Oracle provide several other types of purchase orders to enable longer term planning of organisational spending; releases are cut against such documents as actual procurement is required.

When goods or services are delivered by the supplier to the organisation, they can be *receipted* in the system. The value of this step is to ensure via the 3 way matching process that nothing is paid for until it has been received by the organisation,. (more on matching later). Oracle provide comprehensive facilities for recording receiving activities, including the physical receipt of the goods by e.g. a loading dock, internal forwarding to the requester, inspection and returns. Most organisations simply record the receipt (or return) of the goods or services.

From a system perspective, *suppliers* are part of Purchasing, however they are typically administered by Finance staff and may therefore be regarded as part of Payables by some organisations. In Release 12, Suppliers are part of Oracle's Trading Community Architecture i.e. they are part of Oracle's Master Data Management strategy i.e. to provide a central repository of non-transactional data supported by tools for data cleansing etc.

Note that even if an organisation does not have a warehouse, parts of Oracle Inventory must be implemented (configured) in order to use Oracle Purchasing.

## Payables

Supplier *Invoices* are entered into Oracle Accounts Payable (AP) and may be matched to existing Purchase Orders. The matching process ensures that previously authorised procurement is routed directly into the payment queue, rather than requiring reauthorisation. Matching may be:

2 way i.e. Invoice must match approved PO amount, or

3 way i.e. as above plus received quantity must also match, or

4 way, i.e. as above plus an inspection must have been recorded.

Most organisations seem to be opt for 3 way matching, ensuring that invoices for previously approved spending will be automatically paid if the goods or services have been received.

Some invoices will inevitably also be processed for which no PO previously existed. Typically this will be the case for utility bills, tax payments etc. Depending on the system configuration, such invoices

will be put on hold by the system and a senior Payables staff member will remove the system hold to indicate that he or she has sighted a signature approving the payment of this invoice.

Another source of invoice is *iExpenses*. This web-based Application allows users to record their expenses and uses workflow to route them for approval to the appropriate manager. Like many web-based Applications, there are a set of concurrent processes which bring these records (i.e. the approved expense reports) into the main Application (in this case the Payables Invoice tables).

*Payments* are the other key data in Payables. From the system's perspective these payments are transaction records which are calculated from the pool of unpaid invoices and are either transmitted as files to the bank i.e. as EFT payment instructions or printed as cheques (i.e. one transaction record per document).

The user who submits the payment process is normally not the same person who enters the invoices. This is an example of the principle of *separation of duties*, which is so important to auditors as it lessens the risk of fraud (i.e. of employees creating payments to themselves!)

## Receivables

The key Application in the Order to Cash process is *Oracle Accounts Receivable (AR)*.

It contains 3 main data stores:

- Customer data i.e. master data about the organisation's client base
- Customer invoices (Transactions) i.e. bills sent by the organisation to customers
- Receipts i.e. records of payments received from customers.

*Customer* records are also now part of Oracle's Trading Community Architecture (TCA) but historically they were owned by Receivables and are still regarded as 'owned' by Receivables. Like Suppliers, they consist of a header record which is shared across Operating Units and Site or Address data which is specific to each Operating Unit. Data about an organisation's client base is a valuable resource, this explains why the theme of Master Data Management is a hot topic.

Oracle call customer invoices *Transactions*. This is probably because in addition to sending bills to customers, Oracle can also record credit memos and other types of transactions using the same window. Many transactions originate in *feeder systems* i.e. Applications like *Order Management* which create transactions / customer invoices and interface them to Receivables to be processed through into the General Ledger.

Transaction accounting is created by an Application level tool called AutoAccounting which allows simple rules to be applied to determine how account values are created. (In Oracle Projects there is a similar tool except that the Projects version is like AR AutoAccounting on steroids...)

Transaction accounting may be complicated by non-standard *revenue recognition* requirements. This is because under certain circumstances, accounting rules specify that even if a bill has been issued to a customer, the organisation may not count the invoice amount in its profit calculations. Oracle provide a rule based revenue recognition engine to create the appropriate accounting treatment for such transactions.

*Receipts* record customer payments and may be either entered manually or imported electronically from banks via the so called LockBox interface or as credit card payment via the iPayments Application (in 11i) or Payments (in 12). Receipts are either *standard*, which means they must be applied (matched) to customer invoices or *Miscellaneous* i.e. unrelated to customer invoices. Receipt accounting is partially driven by the Account Generator (a workflow process to determine account codes).

As is the case with all the sub-ledgers, a concurrent process interfaces accounting information to the General Ledger. Such processes are submitted through the Concurrent Manager and are typically scheduled to run at night on automatic resubmission.

## Cash Management

The Cash Management Application is conceptually positioned between Payables and Receivables and its key function is *bank Reconciliation*.

Payables records payments going out from the organisation and Receivables records receipts for payments coming into the organisation. The Cash Management Application is typically configured to import bank statements electronically and the *AutoReconciliation* process matches lines on the statement to payments in AP and receipts in AR. The process also creates new lines to represent bank fees listed on the statements.

Cash Management also provides *cash forecasting* functionality i.e. predicting future cash inflows and outflows based on unpaid customer invoices and un-invoiced supplier receipts. In recent versions of Oracle, it can also utilise data from other Applications like Payroll and Projects for cash forecasting. Most customers do however not use this functionality.